CORNEAINSIGHT

(INSIGHTFUL ANALYTICS FOR EARLY KERATOCONUS DETECTION)

Group Members

Adil Shahzad (01-131222-007)

Muhammad Usman Ali (01-131222-038)

Supervisor: Engr. Aamir Sohail

A Final Year Project submitted to the Department of Software Engineering, Faculty of Engineering Sciences, Bahria University, Islamabad in the partial fulfillment for the award of degree in Bachelor of Software Engineering

May 2026

FYP COMPLETION CERTIFICATE

Student Name: M Usman Ali Enrolment No: 01-131222-038
Student Name: Adil Shahzad Enrolment No: 01-131222-007
Programme of Study: Bachelor of Software Engineering
Project Title: CorneaInsight

It is to certify that the above students' project has been completed to my satisfaction and to my belief, its standard is appropriate for submission for evaluation. I have also conducted plagiarism test of this thesis using HEC prescribed software and found similarity index at 8% that is within the permissible limit set by the HEC. I have also found the thesis in a format recognized by the department.

Supervisor's Signature: ______________________

Date: 15 April 2026 Name: Engr. Aamir Sohail

CERTIFICATE OF ORIGINALITY

This is certify that the intellectual contents of the project **CorneaInsight** are the product of my/our own work except, as cited properly and accurately in the acknowledgements and references, the material taken from such sources as research journals, books, internet, etc. solely to support, elaborate, compare, extend and/or implement the earlier work. Further, this work has not been submitted by me/us previously for any degree, nor it shall be submitted by me/us in the future for obtaining any degree from this University, or any other university or institution. The incorrectness of this information, if proved at any stage, shall authorities the University to cancel my/our degree.

Name of the Student: M Usman Ali

Signature: ________________________ Date: 15 April 2026

Name of the Student: Adil Shahzad

Signature: ________________________ Date: 15 April 2026

PROJECT TITLE (MENTION PROJECT TITLE HERE)

Sustainable Development Goals

(Please tick the relevant SDG(s) linked with FYDP)

SDG No	Description of SDG	SDG No	Description of SDG
SDG 1	No Poverty	**SDG 9**	**Industry, Innovation, and Infrastructure** ✓
SDG 2	Zero Hunger	SDG 10	Reduced Inequalities
SDG 3	**Good Health and Well Being** ✓	SDG 11	Sustainable Cities and Communities
SDG 4	**Quality Education** ✓	SDG 12	Responsible Consumption and Production
SDG 5	Gender Equality	SDG 13	Climate Change
SDG 6	Clean Water and Sanitation	SDG 14	Life Below Water
SDG 7	Affordable and Clean Energy	SDG 15	Life on Land
SDG 8	Decent Work and Economic Growth	SDG 16	Peace, Justice and Strong Institutions
		SDG 17	Partnerships for the Goals

Range of Complex Problem Solving			
	Attribute	Complex Problem	
1	**Range of conflicting requirements**	**Involve wide-ranging or conflicting technical, engineering and other issues.**	✓
2	Depth of analysis required	Have no obvious solution and require abstract thinking, originality in analysis to formulate suitable models.	
3	**Depth of knowledge required**	**Requires research-based knowledge much of which is at, or informed by, the forefront of the professional discipline and which allows a fundamentals-based, first principles analytical approach.**	✓
4	**Familiarity of issues**	**Involve infrequently encountered issues**	
5	Extent of applicable codes	Are outside problems encompassed by standards and codes of practice for professional engineering.	
6	**Extent of stakeholder involvement and level of conflicting requirements**	**Involve diverse groups of stakeholders with widely varying needs.**	✓
7	**Consequences**	**Have significant consequences in a range of contexts.**	✓
8	Interdependence	Are high level problems including many component parts or sub-problems	
Range of Complex Problem Activities			
	Attribute	**Complex Activities**	
1	**Range of resources**	**Involve the use of diverse resources (and for this purpose, resources include people, money, equipment, materials, information and technologies).**	✓
2	Level of interaction	Require resolution of significant problems arising from interactions between wide ranging and conflicting technical, engineering or other issues.	
3	Innovation	Involve creative use of engineering principles and research-based knowledge in novel ways.	
4	Consequences to society and the environment	Have significant consequences in a range of contexts, characterized by difficulty of prediction and mitigation.	
5	Familiarity	Can extend beyond previous experiences by applying principles-based approaches.	

Abstract

CorneaInsight *(Insightful Analytics for early keratoconus Detection),, gives a reliable tool to ophthalmologists for detecting SKC in order to avoid the development of such a dangerous eye condition as keratoconus. In particular, it is really vital to diagnose SKC timely and advise patients on how to protect their health before any interventions with their eyes take place. In other words, KEDS can be considered a "smart second opinion." In fact, this system takes into account two aspects at once, namely objective information (for example, specific measurements of the eye) and subjective information (eye images).*

The algorithm works by using a two-stage AI approach that consists of two methodologies in order to provide more accurate results. Firstly, the methodology referred to as EfficientNet-B0 is focused on pattern recognition within eye images, whereas the other part of the model is called ExtraTreesClassifier and focuses on tabular data processing. Integrating these two methodologies into one framework helps to produce a single outcome on whether the eyes in question are healthy or are diagnosed with Keratoconus. In order to make healthcare professionals trust the outcome of the predictions, the proposed algorithm provides the reasoning behind the decisions made.

To bridge the gap between complex algorithmic processing and clinical utility, the framework integrates dedicated Explainable AI (XAI) mechanisms. This ensuring that the system does not operate as a standard "black box," but instead generates transparent decision pathways alongside its classification. By exposing the explicit features and visual markers driving the diagnostic output, the algorithm provides medical professionals with the necessary interpretability to confidently audit, validate, and trust the system's conclusions in a clinical setting.

Keywords: Keratoconus, Early Detection, Hybrid AI, EfficientNet-B0, ExtraTreesClassifier , Grad-CAM, Clinical Decision Support, Multi-modal Fusion.

Dedication

We dedicate this Final Year Project to our parents. The support, sacrifice, and prayers of both of our parents are the cornerstone of our achievements in life. The unwavering support that you have constantly provided us with is what helped us endure each challenging aspect of this project.

Our hardworking teachers and mentors from Bahria University, this Final Year Project is dedicated to all of you. Without your invaluable advice and technical knowledge, our understanding of Software Engineering would be vastly different. To all our friends, thank you for providing the motivation and support when we needed it most.

Finally, we dedicate this project to everyone that helped us learn and achieve success in the technological world.

Acknowledgments

First and foremost, all gratitude and appreciation must be given to Allah SWT for enabling us with the determination, perseverance, and capacity to complete our Final Year Project successfully.

We wish to thank from the bottom of our hearts our supervisor, Dr. Tamim Ahmed Khan, for his immense guidance, support, and advice in our efforts to improve and enhance our work. His valuable advice and encouragement played a vital role in developing the quality of our project.

We also thank the staff members at Bahria University for their contribution towards imparting knowledge and creating such a favorable learning environment for us.

Our parents' continuous prayers and support, along with the sacrifices that they have made for us, have been our main source of motivation throughout the development of this project. We also express our gratitude to our colleagues who have assisted us in various ways during the process of this development.

Lastly, we thank all those who helped us achieve success in our endeavor, either directly or indirectly.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

The CorneaInsight application is one-of-a-kind in the sense that it helps medical professionals make decisions during diagnosis. It has been enabled through its advanced technology to be able to detect a specific type of eye disease called Keratoconus. This is vital because detecting signs of Keratoconus will help save live, especially when patients have their vision corrected surgically.

1.1. Motivation

Keratoconus diagnosis, especially the hidden variant called "subclinical," is still a difficult task for ophthalmologists. Usually, doctors have to study many data at once, both numerical data and complex eye image, to discover any signs of disorders. In some instances, such a complicated process can lead to mistakes in diagnos due to ambiguity.

The necessity for such a system is obvious. Hence, the development of KEDS came into being. This system is analogous to a smart second-opinion system used by ophthalmologists since they use various inputs. By analysing the information along with the images of the retina, the accuracy of the system are increased. The most crucial aspect is that the decision made by KEDS will be identical, regardless of whether the information is verified by another person and analysed elsewhere.

1.2. Objectives

The primary objective of the project involves designing a smart computer program that will help the ophthalmologist work efficiently. The following are some of the objectives that we hope to achieve through this project:

- **Designing a Multi-modal Diagnosis Engine:**

  Here, the objective involves designing a double modal Artificial Intelligence engine that takes into account two variables at once, which include structured clinical images and visual eye images. The evaluation of these two variables helps in making an accurate diagnosis.

- **Engineering of a Unified Cross-Platform Interface**:

  We are planning on creating a universal app that will work perfectly on any device. With the use of software like Flutter and Electron.js, we can ensure a perfect performance of the application on any device.

- **Implementation of Explainable AI (XAI) for Clincal Trust**:

  We should not overlook the confidence of the doctor regarding the results provided by the AI technology. The heat maps will ensure that the areas pointed out by the algorithm are highlighted to understand how the algorithm decided whether the eye map is healthy or not.

- **Establishment of a Secure Clinical Data Model**:

  Supabase will be included in our system as this architecture will help in making sure that the patient information are stored safely and correctly. The healthcare professionals will be able to keep a check on their patients' status to know whether there are any changes or not.

1.3. Main contributions

KEDS, which may also be referred to as Cornea Insight, represents a revolution in how testing was done in the past through it use of more sophisticated technology as compared to the methods employed in the past.

Through KEDS, there is provision of an end-to-end solution involving deep learning imaging and clinical data in one single interface. Whereas, in the past, doctors had to carry out manual test to obtain information from various sources, the same has become possible now through the delegation of tasks through the workflow process.

In conjunction with the most effective artificial intelligence techniques like EfficientNet-B0 and ExtraTreesClassifier in diagnosing diseases, KEDS analyzes patients' information by considering a variety of different factor. Furthermore, the use of technology ensures that personal data is kept private because it gives managers an opportunity to regulate access to this data and the possibility to conduct diagnoses on patients.

The medical professionals will have the chance to look at the outcomes obtained with the help of artificial intelligence in both visual form and in the form of a list of related variable. This allows health practitioners to make better-informed decisions concerning patients' treatment and surgeries.

1.3.1. Existing Systems

A considerable portion of the latest technologies used for eye tests is embedded in the device itself, and there is no interaction between various device. Furthermore, doctors have to analyze visual maps separately and correlate them with clinical figures themselves.

Most obsolete computer programs employ simplistic algorithms for the diagnosis of ocular disorder; however, their efficiency is very low due to the possibility of overlooking the early signs of keratoconus caused by their incapability to identify small morphological changes, which cannot be estimated by modern methods of testing. Besides, the old software does not account for patient history or explain the reasons behind the computations. Consequently, doctors are unable to trace the development of the disease.

1.3.2. Proposed System

CorneaInsight overcomes these challenges with a detailed and innovative clinical solution:

- **Unified Platform**:
  The system will provide a one-stop solution whereby doctors can easily view and analyze eye images and clinical parameters without any issues.

- **Hybrid Core**:
  Both eye images and clinical information will be analyzed by two powerful AI models. The use of two powerful AI models increases the chances of detecting eye diseases at their early stages.

- **Role-Based Control**:
  Roles will be allocated to individuals who are involved in the whole process. For instance, the technicians will enter the data while eye experts will handle the diagnosis process. Patients will be allowed to view their clinical analysis.

- **Explainable AI (XAI)**:
  Powerful visualization technologies will be used to analyze eye images, which will lead to the generation of image representing portions of eye images that are used by AI to make decisions.

1.3.3. Workflow Overview

CorneaInsight is structured to ensure a smooth clinical procedure. The following workflow outlines the step-by-step procedure followed within the system:

Workflow Steps:

1. **Technician/Clinician Inputs Data**: Enters numerical metrics & uploads corneal map JPEGs.
2. **AI Engine Processes Data**: The smart AI system analyzes the collected information and diagnoses by analyzing visualize images.
3. **System Saves Record**: In this process, the patient's data and AI diagnostic results are stored in the cloud.
4. **Clinician Reviews Results**: The eye doctor analyzes the results generated by the smart AI system.
5. **Report Generation**: The last process in this workflow is report generation in PDF form.

Workflow Diagram:

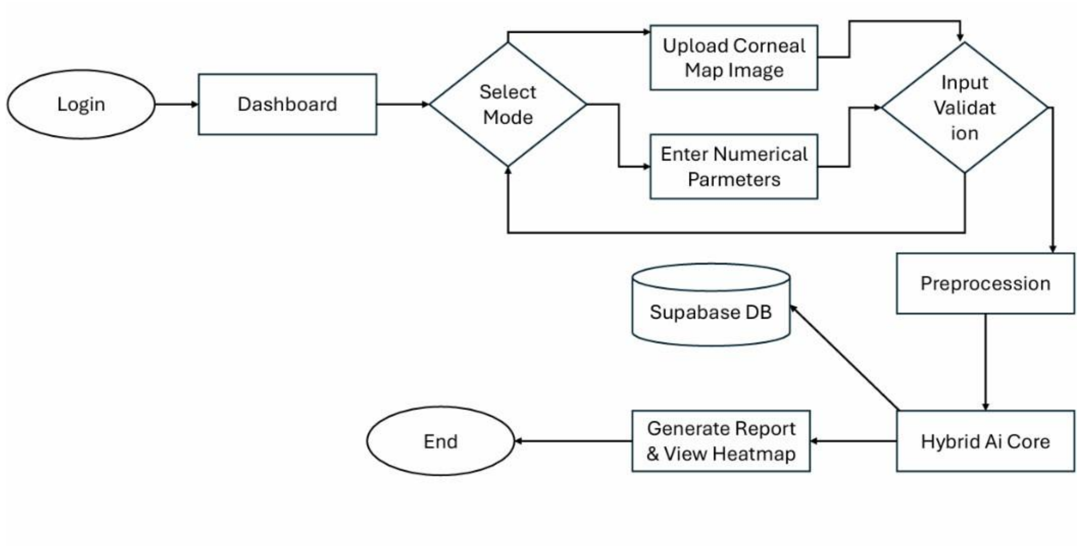


Figure 1.1: *Workflow*

This figure 1.1 workflow illustrates the end-to-end user journey of the application, starting from secure authentication and progressing through a multimodal data entry stage where users can upload corneal images or enter numerical parameters. Following input validation and preprocessing, the data is processed by a Hybrid AI Core to generate diagnostic reports and heatmaps, with all results simultaneously archived in a Supabase DB for record-keeping.

1.4. Report organisation

Following the structure of the provided example, here are the Report Organization for your KEDS (CorneaInsight) project:
This report is structured as follows:

Chapter 2: Literature Review - In this chapter, we describes existing ways of Keratoconus detection, existing medical imaging methods, and advances in artificial intelligence.

Chapter 3: System Requirements - In this chapter, we provides description of our design methodology, technologies we use in our system (Flutter and Electron.js), and design considerations.

Chapter 4: System Design - In this chapter, we provide description of our design methodology, technologies we use in our system (Flutter and Electron.js), and design considerations.

Chapter 5: Implementation - In this chapter, we provides an overview of implementation process of the proposed AI hybrid model and implementation of other software components (for instance, Supabase database).

Chapter 6: Testing and Evaluation - In this chapter, we pays attention to testing procedure, performance analysis of our AI and ways to provide correct system performance in a medical context.

Chapter 7: Conclusion and Future Work - In this chapter, we summarize our research concerning the problem and point at future direction for software improvement.

Chapter 2

Background Study/Literature Review

The early diagnosis of eye-related problems is essential for preventing vision loss and becoming blind. Some of the conditions cannot be detected using any clinical observation methods. For example, subclinical keratoconus extremely hard to diagnose. This create serious difficulties for patients wishing to conduct vision correction surgery. CorneaInsight is a state-of-the-art software application designed specifically for medical specialists. It combines two approaches, one of which uses the modern visual analysis of the collected data and metrics.

CorneaInsight is a software application that is meant to identify any defects that might be present in the eye by analyzing special images taken of the eye, as well as other medical information regarding the eye, and determining whether there are any Keratoconus or weak point in the cornea. This chapter will give an overview of eye test theories and examine currently available devices for the medical industry to prove how KEDS is better than current technology.

2.1. Significance of Corneal Analysis

Corneal screening is one of the key factors that aid doctors in diagnosing any diseases present, their stage of progress as well as eligibility of the patient to have surgery done. Ophthalmology screening now is mainly performed through one of the two techniques:

- **Numerical Parameter Analysis:**
  In this case, there is the use of some numerical parameters, including the maximum curvature and thinnest area that are analyzed without considering any visual images. This is very effective, especially in the late stage of disease when the pre-determined threshold values have been attained. However, this method tends to miss out on the hidden cases which cannot be detected from the numerically calculated values.

- **Topographic Image Analysis:**
  Here, the eye map is used to detect abnormal patterns in real time through analysis of the geometric and curvatures of the eye. Through this method, the doctor can follow up on the development of the eye to detect any risk factors. This method is quite computational.

2.2. Significance of Corneal Analysis

2.2.1. Hybrid AI Multi-Modal Core

The CorneaInsight uses the Hybrid AI Core to improve the efficiency of several detection engines. The process involves the rapid analysis of the data in the image using the EfficientNet-B0 algorithm in detecting patterns and the ExtraTreesClassifier technique in examining the structured clinical data.

2.2.2. Explainable AI (XAI) and Grad-CAM

In KEDS, rational visual integration is used to increase clinical confidence. Grad-CAM heatmap generation from the KEDS highlights some features in the cornea maps. The assurance is provided by the algorithm that the diagnosis will be based on these features only. Thereby, no errors or misdiagnoses of black box types can occur.

2.3. Key Technologies Utilized in CorneaInsight

2.3.1. Role-Based Access Control (RBAC)

The CorneaInsight application Implement RBAC architecture to secure the confidentiality and integrity of the delicate ophthalmologic information. Below is a hierarchy of the access levels based on the user role:

- **Ophthalmologists (Doctor)**:
  Have higher access privileges enabling them to conduct a diagnosis through hybrid means by applying AI technology, view all patients' data, and generate PDF reports of each patient.

- **Support Staff/Technicians**:
  Having restricted access rights allowing them to perform administrative operations such as inputting numbers and uploading images. They cannot validate the diagnostic results.

- **Patients**:
  Are granted restricted access rights to allow them to monitor their health status and download reports.

This hierarchical structure minimizes the risk of unauthorized data exposure and ensures that clinical operations remain streamlined and accountable.

2.3.2. Backend AI API and Model Integration

CorneaInsight relies on an advanced RESTful API, implemented using a highly scalable backend coded in Python/FastAPI. In this regard, all complex calculations can be offloaded from the user's device to the cloud.

- **Multi-Modal Processing**:
  Specifically, the API analyzes the EfficientNet-B0 neural network for images and the ExtraTreesClassifier algorithm for numeric data concurrently. Moreover, the API standardizes the values of the clinical parameters. Lastly, it converts the images in JPEG format into tensors (224*224*3).

- **Ensemble Synthesis**:

    The backend implements a meta-model that combines the soft predictions produced by both the AI components. On their basis, the API calculates the final Classification and Confidence score.

- **Explainability Generation**:

    Meanwhile, the backend executes the Grad-CAM process to produce a heatmap mask. This is transmitted to the front-end for the visual representation of clinical reasoning.

2.3.3. Cloud Persistence and Synchronization

CorneaInsight uses Supabase (Managed PostgreSQL) as the database engine for secure and real-time data storage.

- **Longitudinal Tracking**:

    The database model enables the retention of previous records regarding medical diagnosis. The system determines if there is any improvement or deterioration of the illness depending on the continuous checks performed on the patient.

- **Data Security**:

    All messages sent from the patient to the Supabase server are transmitted in an encrypted form using TLS 1.3. Furthermore, all patients' health information is protected using AES-256 encryption.

2.4. Existing Diagnostic Systems

There are several known systems for identifying and reporting Keratoconus. We use these systems in this section to learn about their strengths and areas where CorneaInsight can improve.

2.4.1. Pentacam HR (Belin/Ambrósio Enhanced Ectasia)

Pentacam is an extensively utilized hardware service that uses Scheimpflug technology for collecting tomography information.

- **Capabilities**:

    It possesses great diagnostic ability to detect severe cases of KC because of its distinctive D-index function. Being widely used in hospitals, it becomes the gold standard for quick diagnosis.

- **Limitations**:

    Its ability to diagnose subclinical cases may be limited due to its thresholding system. Also, it lacks flexibility features because of its incapability of including behavioral analysis based on deep learning technology.

2.4.2. Standard AI Research Models

Some CNN models exist for research purposes in corneal classification.

- **Capabilities**:
  They contain deep learning features and show excellent accuracy on specific datasets.

- **Limitations**:
  They show high accuracy levels, yet suffer from a black-box problem, setting them apart from CorneaInsight. They lack an interface and are not suitable for clinical use.

2.5. Methods

2.5.1. Corneal Analysis Methods

CorneaInsight uses several automated tools for assessing the condition of the cornea:

- **Image Pattern Recognition**:
  Machine learning algorithms are deployed to recognize regions with protrusion, thinning, and asymmetry of the corneal plots of elevation and curvature.

- **Numerical Index Benchmarking**:
  It does the comparison of some significant clinical indices like K_{max} and thinnest pachymetry against medical truth.

- **Explainable AI (XAI) Mapping**:
  Grad-CAM algorithm is applied to the output of the CNN model to generate a heatmap of the regions of the corneal plots responsible for AI detection decisions.

2.5.2. Development & Integration Methods

- **Cross-Platform Engineering**:
  This application uses cross-platform engineering by taking advantage of Flutter and Electron.js to have a single code base that works with mobile and desktop platforms such as Windows, macOS, Android, and iOS.

- **API-Driven Inference**:
  Models used within this project are based on an API inference methodology. The use of FastAPI Server is implemented within this project to infer models. It allows us to move some complex operations such as deep

learning inference into the back end to ensure performance even while conducting inference with low-end client devices.

- **Role-Based Access Control (RBAC)**:
  This solution uses the RBAC access model to manage different user accounts, namely "Doctor," "Patient," and "Technician" ones. This helps maintain secure access to certain functions and data.

2.6. Comparative Analysis of CorneaInsight with Existing Tools

Feature	**Pentacam Native**	**Research CNNs**	**CorneaInsight (KEDS)**
Hybrid Analysis	No	No	Yes
Visual Rationale (XAI)	No	No	Yes
Cloud Persistence	No	No	Yes
Cross-Platform	No	No	Yes

Table 2.1: *Comparative Analysis of CorneaInsight with Existing Tools*

CorneaInsight makes itself unique by offering a fully integrated hybrid workflow combined with cloud-based longitudinal monitoring and visual report generation.

2.7. Summary

CorneaInsight technology can help to enhance the simplicity in acquiring a correct diagnosis by merging the conventional methods of eye testing with its advanced diagnostic methods. The incorporation of the hybrid processor ensures the accuracy of the diagnostic process. The aim of CorneaInsight is to overcome the limitations of eye-screening machines that are hard to interpret.

Chapter 3

System Requirements

KEDS provides a set of fully optimized functionalities which are expected to offer reliable and objective support during the diagnosis process. Below is a list of functionalities in the product classified in accordance with their functions. Every functionality provided depends on its name. We will also provide a description of how each function is used.

3.1. Use Case Diagram

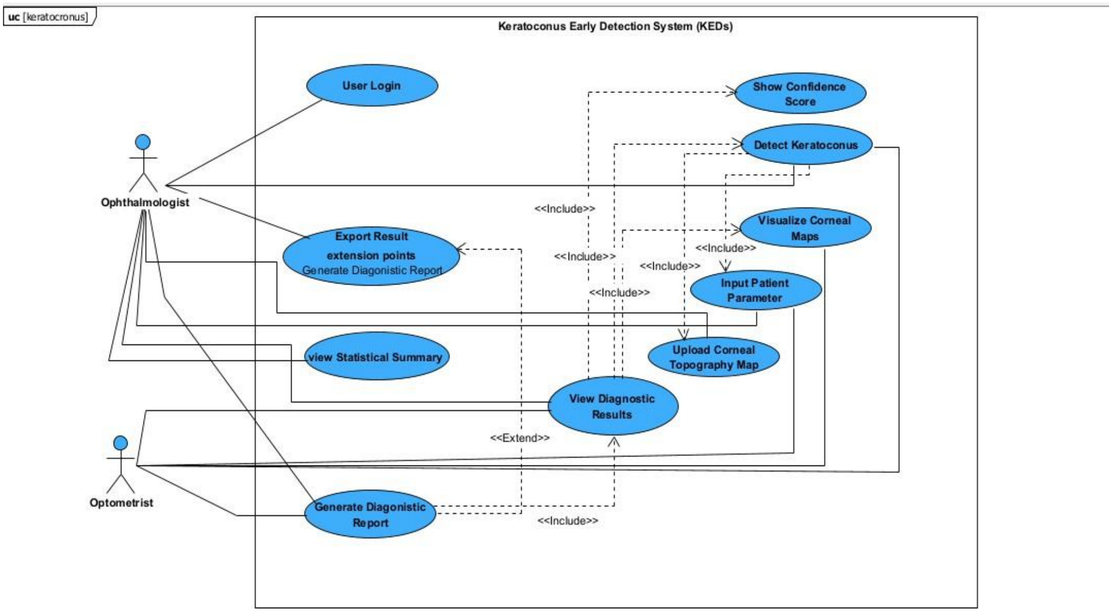


Figure 3.1: *Use Case Diagram*

This usecase diagram outlines the functional interactions between the Ophthalmologist and Optometrist actors within the CorneaInsight system, specifically highlighting their roles in data input and diagnostic analysis. It showcases how viewing diagnostic results triggers essential sub-processes like detecting Keratoconus, visualizing corneal maps, and generating comprehensive reports for clinical decision-making.

3.1.1. User Login

This feature provides all capabilities necessary for clinicians to securely access and maintain their identity within the CorneaInsight platform, ensuring data privacy and proper credential validation.

Use Case ID:	**FR-1.1**	
Use Case Name:	**Clinician Authentication (Login)**	
Actor(s):	Corneal Specialist, General Clinician, System Administrator	
Pre-Conditions:	The user must have a registered account and the system must be online and connected to **Supabase**.	
Priority:	High	
Basic Flow:		
Actor Actions		**System Response**
1	User enters username (email) and password in the relevant fields.	**2** System checks the provided credentials against the secure **Supabase** database.
3	User clicks the "Login" button.	**4** If credentials are valid, the user is authenticated, and the **Dashboard Screen** is displayed.
Alternative Course of Action (if any)		
Actor Action		**System Response**
		4.a If credentials are invalid, an error message is displayed: "Invalid email or password. Please try again **4.b** If there are 3 consecutive unsuccessful attempts, access for that user is temporarily blocked for 1 hour for security.

Table 3.1: *user and session management*

3.1.2. Result Visualization

This feature is the core value proposition of CorneaInsight. It defines the parallel execution of the hybrid AI models, the fusion of their outputs, and the detailed presentation of the final diagnostic outcome and rationale to the clinician.

Use Case: Generate AI Diagnosis and Rationale

Use Case ID:	**FR-3.1**
Use Case Name:	**Generate AI Diagnosis and Rationale**
Actor(s):	Corneal Specialist, General Clinician
Pre-Conditions:	1. The Clinician must be authenticated. 2. All multi-modal data has been successfully submitted via FR-2.1. 3. The Backend AI API must be operational.
Priority:	High
Basic Flow:	

Actor Actions		System Response
1	Clinician initiates processing (explicitly or after submission).	**2 S**ystem displays a "Processing Data" loading indicator.
		3 Backend receives the request, normalizes numerical features, and converts image JPEGs to tensors (224x224x3).
		4. Backend executes the ExtraTreesClassifier (Tabular) and EfficientNet B0 (Image) models concurrently.
		5.The Ensemble (Meta-Model) receives both soft probability outputs and computes the Final Classification and Confidence Score.
		6.The Backend generates the Grad-CAM heatmap mask based on the EfficientNet B0 output.
		7.System securely saves the result and patient metadata to Supabase (FR-4.1).
8	System returns the Final Classification, Confidence, and Grad-CAM mask data to the **Flutter** frontend	**9.** Frontend displays the comprehensive "Results & Visualization" screen (FR-3.2).
Alternative Course of Action (if any)		
Actor Action		**System Response**
		4.a If the Backend AI API fails to respond (e.g., model load error or service timeout). **4.a.i** System displays a clear error indicating "Diagnostic Service Unavailable" and logs the error details for System Administrator review.

Table 3.1: *Generate AI Diagnostic and Rationale*

3.1.3. Reporting and Longitudinal Monitoring

This feature provides the capability to retrieve, view, and generate permanent documentation of past diagnostic results, supporting the clinical requirement for tracking disease stability or progression over time**.**

Use Case: Generate Diagnostic Report (PDF)

Use Case ID:	**FR-4.3**	
Use Case Name:	**Generate Diagnostic Report (PDF)**	
Actor(s):	Corneal Specialist, General Clinician, Optometrist	
Pre-Conditions:	The patient must have at least one finalized diagnostic result stored in **Supabase**.	
Priority:	High	
Basic Flow:		
Actor Actions		**System Response**
1	Clinician navigates to the patient's history page.	**2** System retrieves the list of past diagnostic results from Supabase.
3	Clinician selects a specific result record and clicks the "Generate PDF Report" button.	**4** System packages the patient data, classification, confidence score, and visual rationale (Grad-CAM image) into the standardized PDF report format.
		5 System presents the finalized PDF report file to the user.
6	Clinician downloads or prints the PDF report.	**7** System logs the report generation event for auditing purposes.
Alternative Course of Action (if any)		
Actor Action		**System Response**
		4.a If the report generation fails (e.g., rendering error). **4.a.i** System displays an error: "Unable to generate report. Please check the data integrity and try again."

Table 3.1: *Reporting and Monitoring*

3.1.4. Specialized Single-Modal Analysis

This feature defines the ability to run independent classification using only one modality of data. This capability is primarily included for model testing, validation, and specialized clinical use cases where a full data set may not be available.

Use Case: Image-Only Detection

Use Case ID:	FR-4.4	
Use Case Name:	**Image-Only Detection**	
Actor(s):	System Administrator (Testing), Corneal Specialist (Special Case)	
Pre-Conditions:	1. User is authenticated. 2. A minimum of one required corneal map image (JPEG) has been uploaded.	
Priority:	Medium	
Basic Flow:		
Actor Actions		**System Response**
1	User selects the "Run Image-Only Analysis" option.	**2.**System packages the image tensors and sends the request to the Backend.
		3.Backend routes the data directly to the EfficientNet B0 component, by passing the ExtraTreesClassifier step.
		4.EfficientNet B0 returns the Image-Only Classification and Confidence Score.
		5.System saves the result (marked as "Image-Only") to Supabase.
6	System returns the result to the Frontend.	**7.**Frontend displays the result, clearly labeled as "Image-Only Prediction."
Alternative Course of Action (if any)		
Actor Action		**System Response**
	4.a The image is corrupted or missing key features.	**4.a.i** System returns an error: "Image-Only analysis failed due to poor image quality."

Table 3.1: *Image only detection*

Use Case: Numerical-Only Detection

Use Case ID:	FR-4.5
Use Case Name:	**Parameter-Based Progressive Monitoring**
Actor(s):	Corneal Specialist, General Clinician
Pre-Conditions:	1. User is authenticated. 2. The patient has at least two separate diagnostic records saved in Supabase.
Priority:	High

Basic Flow:		
Actor Actions		**System Response**
1	Clinician navigates to the patient's history.	**2** System retrieves all available historical records and associated numerical features from Supabase.
3	Clinician selects two or more specific records (visits) for comparison.	**4** System computes and displays the rate of change for key progressive metrics (e.g., change in K_{max} and pachymetry over time) between the selected visits.
		5 System presents a Progressive Monitoring Chart (NFR-4.3) showing the trend of these key metrics across the visits, overlaid with clinical thresholds.
Alternative Course of Action (if any)		
Actor Action		**System Response**
		4.a If fewer than two historical records are available for the patient. **4.a.i** System displays an error: "Insufficient data for progressive monitoring. Requires at least two historical records."

Table 3.2: *Numerical only detection*

3.2. Functional Requirements

The functional requirements define the specific behaviors and services that the KEDS platform provides to medical professionals to facilitate early keratoconus detection.

3.2.1. User and Session Management

- **FR-1.1: Clinician Authentication:**

  Clinician authentication to use the system is possible through Supabase Auth, where users provide their registered emails and passwords.
- **FR-1.2: Role-Based Access Control (RBAC):**

  Access to the diagnostic tools of the system is controlled by roles assigned to the users. Users can have the following roles: Ophthalmologist, Technician, or Patient. It is important for ensuring that patients' information remains secure from any unauthorized use.
- **FR-1.3: Secure Logout:**

  A logout feature is available within the platform. The operation will log out the user and erase all patients' data from the working memory.

3.2.2. Multi-modal Data Ingestion

- **FR-2.1: Numerical Parameter Entry:**
  Clinicians can input numerical values of K_{max}, ISV, IVA, and thinnest pachymetry parameters through an intuitive interface.

- **FR-2.2: Topography Map Upload:**
  Different types of corneal topography maps (7 types), which can either be elevation or curvature maps captured by eye imaging devices, are uploaded by clinicians as JPEG/PNG formats.

- **FR-2.3: Data Validation:**
  Validation of entered data will be conducted on-the-go, that is before processing the data.

3.2.3. Intelligent Diagnostic Processing

- **FR-3.1: Hybrid AI Analysis:**
  The system launches the EfficientNet-B0 image-based model and the ExtraTreesClassifier numerical classifier at once after getting the input data.

- **FR-3.2: Multi-Modal Fusion:**
  The AI Core unites the output of both classifiers into a common probabilistic classification label, e.g., Normal or Keratoconus.

- **FR-3.3: Visual Rationale Generation:**

  The system creates the Grad-CAM overlay heatmap for all image-based predictions. The heatmap points out certain areas of diagnosis on corneal topography images.

3.2.4. Reporting and Longitudinal Monitoring

- **FR-4.1: Secure Data Persistence:**
  The system persists all diagnoses into the database called Supabase automatically.
- **FR-4.2: Historical Record Retrieval:**

  Access to the patient's previous diagnostics records is available within the system. The records can be utilized to monitor disease development.

- **FR-4.3: Standardized PDF Generation:**
  The system automatically generates the official report document in PDF format with the diagnosis, probabilities, visual rationales, and patient's details.

- **FR-4.4: Progressive Monitoring Chart:**

  The system visualizes the progress of the various parameters, such as K_{max}, through time by comparison between visits.

3.3. Non-Functional Requirements

In this part, the qualities of the developed software, performance metrics and critical limitations that the KEDS must satisfy are discussed. All of these aspects determine the functionality of the software application and are critical for it to be effective and comply with medical regulations.

3.3.1. Performance Requirements

- **NFR-1.1:**

  The total processing time of the AI Diagnostic Processing (FR-3.1) should not be more than 5 seconds. It can be defined as the period from sending the request and processing to receiving results and Grad-CAM output from the patient.

- **NFR-1.2:**

  The system should manage up to 20 diagnostic requests simultaneously in 5 seconds.

- **NFR-1.3:**

  The Flutter app should load the Dashboard screen in 2 seconds after successful logging in.

3.3.2. Safety Requirements

- **NFR-2.1:**

  The system shall offer a notification signifying that the diagnostic support tool under use is KEDS. In addition, the system should state that the diagnostic support tool should never replace the doctor's judgment in making his decisions.

- **NFR-2.2:**

  All significant numbers are subjected to range checking. Range checking would help eliminate any miscalculations made because of errors in utilizing inappropriate numbers.

- **NFR-2.3:**

  The hybrid artificial intelligence core must use versioning. Versioning allows tracking the diagnoses made to the specific version of the model generating those diagnoses.

3.3.3. Security Requirements

- **NFR-3.1:**

  All patient records and personal health information (PHI) stored in Supabase must be protected using industry-standard encryption. This includes data at rest and data in transit via HTTPS/TLS 1.2+.

- **NFR-3.2**:

The application should support a password management policy for the Clinician's account. It will require at least a 10-character minimum password, with an alphanumeric character combination.

- **NFR-3.3:**

Role-based access control will be enforced in the application. The Ophthalmic Technicians will not have access to the Model Version updates or system administration capabilities.

3.3.4. Software Quality Attribute

- **NFR-4.1:**

The user interface (Flutter) needs to score equal or greater SUS scores of 75 points when evaluated by an appropriate sample of medical professionals.

- **NFR-4.2:**

When filling data forms, there will be the inclusion of visual aids as well as tooltips. This is meant to ensure that the user can understand the medical range of all numeric parameters.

- **NFR-4.3:**

For the feature "Monitoring," patient history and progression data need to load in 3 seconds.

- **NFR-4.4:**

The availability of the back-end API will be 99.9% throughout the specified operating period. However, planned maintenance will not be counted towards this metric.

- **NFR-4.5:**

Corruption percentage of the data saved using Supabase must be below 0.01%.

- **NFR-4.6:**

All code used for the Flutter front-end and Python back-end should have well-structured coding guidelines such as linter. Commenting on everything extensively should ensure that even a fresh developer can comprehend how the logic works after only two weeks of learning.

- **FR-4.7:**

Back-end AI core should remain modular in nature, ensuring that a better-performing version can be utilized, i.e., from B0 to another model. The change should not require modifications to the way flutter front-end sends information.

- **NFR-4.8:**

Any changes to the API Guide documentation concerning the back-end endpoints or returned data should be made within 48 hours.

- **NFR-4.9:**

It must be consistent in function within various Web browsers like Google Chrome, Mozilla Firefox, and Safari, as well as in Mobile platforms like those in iOS and Android. The creation of this system is made possible by using the Flutter technology without having different codes.

- **NFR-4.10:**

The output of the data generated by the solution must come in standardized formats such as in PDF documents which could be printed and even incorporated into the EHR.

- **NFR-4.11:**

The frontend must be designed in such a way that it could automatically adapt according to its design and size. This must allow it to work from smartphones up to desktops.

3.3.5. Business Rules

- **NFR-5.1 (Diagnosis Finality):**

After recording any diagnosis, only an authorized Corneal Specialist / Ophthalmologist can change the data in a patient's record.

- **NFR-5.2 (Ownership and Access Control):**

Only the clinician who generated the patient record or to whom sharing access to this record was provided will have access rights and ownership.

- **NFR-5.3 (Update of Model):**

Any change to the model used in the production environment should be validated before applying it. The measures AUC and Sensitivity need to be checked using a hold-out test set.

3.4. Interface Requirements

The software includes one UI that is designed according to Material design philosophy. The color palette includes blue color for medical applications and white. The UI allows for a responsive layout that works well with the resolution of 1920x1080 screens and new mobile devices' resolution.

3.4.1. User Interface

The front end (Flutter/Electron) UI interfaces with the AI Diagnostic Engine using RESTful API supported by FastAPI platform. All communication uses JSON protocol for the control commands. Images with high resolution are sent in Base64 and Multipart-Form protocols.

3.4.2. Software Interface

The front end (Flutter/Electron) UI interfaces with the AI Diagnostic Engine using RESTful API supported by FastAPI platform. All communication uses JSON protocol for the control commands. Images with high resolution are sent in Base64 and Multipart-Form protocols.

3.4.3. Communication Interface

All communication in this system happens through TLS 1.3 secure connection using the HTTPS protocol. The interface follows the optimistic user interface philosophy, where all the information is stored locally before being synchronized with the cloud database.

3.5. Database Requirements

The data layer is engineered to handle complex relational medical records with high performance and security.

3.5.1. Storage Technology

The program uses Supabase (PostgreSQL) ensuring safe cloud-based storage of users' profiles and metadata related to the patients.

3.5.2. Relational Schema

The rigorous schema is used here where fields have been defined according to the structure of the Patient Document. Tables that are available in the database include users, patients, diagnostic_report, and ai_predictions.

3.5.3. Performance Optimization

Important fields like clinicianId and patientId are indexed making sure that the retrieval time of the history will not exceed 3 seconds.

3.5.4. Data Integrity

Integrity of the data has been ensured by reducing the percentage of corrupted data under 0.01% in the system. In addition to that, the system performs database encryption in disk using AES-256 algorithm.

3.6. Project Feasibility

3.6.1. Technical Feasibility

The project is technically feasible as it leverages proven modern frameworks:

- **Deep Learning Backbones:**

  Combining EfficientNet-B0 and ExtraTreesClassifier gives a solid ground to start for multi-modal classification tasks.

- **Cross-Platform Delivery:**

  With Flutter and Electron.js, we can reused our application code on all platforms, which substantially decreases the overhead in development efforts.

- **Hardware Availability:**

  The list of hardware needed for the learning process and its implementation on the device, including NVIDIA with 4GB VRAM, is perfectly covered by modern medical computers.

3.6.2. Operational Feasibility

The system is designed to integrate into existing clinical workflows without significant disruption:

- **Role-Based Workflows:**

  KEDS follows the traditional workflow where "technicians scan, doctors analyze" because we have assigned unique roles for Technicians and Ophthalmologists.

- **User Training:**

  Our software package provides a detailed User Manual (PDF) and AI Model Deployment Guide. It ensures that the users will be able to understand and analyze Grad-CAM heat maps and quantitative reasoning techniques.

- **Accessibility:**

  The functionality of the native app makes the process efficient within the screening environment.

3.6.3. Legal & Ethical Feasibility

KEDS is engineered to comply with international medical software standards:

- **Privacy Compliance:**

  All information handling involving the patient data is done to be HIPAA-compliant. This ensures that all patient health information will remain confidential and protected.

  M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in *Proc. 36th Int. Conf. Mach. Learn. (ICML)*, 2019, pp. 6105–6114.

- **Data Anonymization:**

  Identifying information of the patient is removed from the data set in training and testing sessions.

- **Ethical Disclaimer:**

  There will be an ethics statement as part of the study design that ensures that KEDS will be used only as a "Diagnosis Support Tool." This ensures that ultimate decision-making power is retained by the trained clinician.

3.7. Conclusion

The formulation of the complete requirements specification for the Keratoconus Early Detection System (KEDS) is discussed in this chapter. Both the basic Functional Requirements (FRs) related to the multimodal data acquisition process and the non-functional requirements (NFRs) regarding the computational delay and clinical safety norms are identified here. Through use cases and activity diagrams, the efficiency of the proposed framework are validated for enabling complex clinical procedures necessary for the early detection of sub-clinical keratoconus.

Chapter 4

System Design

This document outlines the precise strategy used to implement the Universal Data Client (UDC), one program that will perform flawlessly both in mobile and desktop versions (for Android/iOS devices, Windows, Mac and Linux). The main focus of the project lies in developing an efficient, lightning-fast application that will be easy to modify and improve in the future by any developer working on this platform. This is facilitated by the development process involving the used of the popular Flutter framework, which allows for writing only once, resulting in the same experience across all types of devices used by users. Apart from that, we used the MVVM approach to ensure that there is an exact separation between the application's user interface and the business logic (or how it is supposed to function). In the case of the desktop version, we rely on the Electron.js framework, which will serve as the basis for communication between the software and the OS.

4.1. Design Approach

In addition, the Keratoconus Early Detection System (KEDS) adopts Layered Client Architecture, together with the MVVM architectural pattern, which guarantees that there will be a clear separation of concerns. This decision is made to make sure that there will be no mingling between the user interface and the complicated business logic contained within the Hybrid AI Core.

- **Presentation Layer (View):**

  The View Layer is developed with the help of Flutter framework in case of mobile app and encapsulated by Electron.js in case of desktop application. The View Layer can be employed for displaying UI and receiving user inputs through observation of reactive streams in ViewModel Layer.

- **Domain Layer (View Model):**

  This Layer is concerned with the management of the application state and business logic. This Layer is responsible for handling user click events and managing the data flow from UI layer to data source layer.

- **Data Layer (Model & Repository):**

  Model/Repository Layer categorizes the data source by following the Repository design pattern. It helps in fetching data from either cloud database with Supabase or artificial intelligence API endpoint with FastAPI service.

4.2. Design Constraints [Optional]

Some factors significantly influence the design of CorneaInsight and ensure the effectiveness of this solution in practice in use for medical professionals.

- **Deep Learning Requirements:**
  The image-based component of diagnostics must rely upon the EfficientNet-B0 deep learning model. This means that all images must be resized to 224*224*3 pixels.

- **Inference Performance:**
  Five seconds can be considered the maximum acceptable time during which the process of inference occurs to not hinder diagnosing the patients

- **Platform Consistency:**
  This software is expected to work equally well under Windows, macOS, Android, and iOS operating systems using one and the same codebase. For this reason, the approach of cross-platform application development using Flutter must be employed.

- **Regulatory & Privacy Standards:**
  Software is developed considering the requirements established by HIPAA. Thus, such technologies as TLS 1.3 and AES-256 are used.

- **Explainability Mandate:**
  It is a basic requirement that all predictions made by KEDS must come with the Grad-CAM explanation, along with the feature importance scores to explain how it came to that conclusion.

- **Hardware Independence:**
  KEDS is a system that can work independently of any hardware and does not have any control over the imaging device. It only uses standard image formats like JPEG and PNG images.

4.3. System Architecture

The Keratoconus Early Detection System (KEDS) utilizes the Three-Tier Layered Architecture approach. It guarantees scalability, security, and modularity to perform AI inference. There are three layers involved in the Keratoconus Early Detection System. They are as follows: the Frontend layer, the Backend layer, and the Data & Intelligence layer.

4.3.1. Frontend Layer (Client-Side)

The presentation tier is engineered as a unified cross-platform interface. It consists of two primary applications:

- **Desktop/Web Application:** Desktop/Web application has been created using Flutter framework along with Electron.js which will provide native functionalities of the platform.
- **Mobile Application:**
  Mobile applications have been created using the native functionality of the Flutter framework for both Android and iOS.

- **Communication:**
  Both the clients will use Restful API to communicate with the server. The protocol used for communication is JSON & multipart form for the images.

4.3.2. Backend Layer (Server-Side)

The logic tier serves as the central hub for the entire system's operations:

- **Core API Framework**:
  Built with the help of Python FastAPI, this layer deals with the fast processing of queries and authentication of users. It further ensures that there is proper exchange of information between UI and the underlying AI modules.

- **Service Coordination**:
  This is where routing of clinical information to AI core takes place. This module also handles the retrieving of patients' history from the database.

4.3.3. Data & Intelligence Layer

The layer includes all the smart components required to secure storage of the information and diagnose it in a smart way:

- **Database (Supabase/PostgreSQL)**:
  The storage place for all data including User Auth Credential, Patient Record, and Scan Meta Data. There are where the report from the saved image analysis report, parameter analysis report, is stored.

- **Hybrid AI Model Core**:
  The intelligence module consists of EfficientNet-B0 for the images and ExtraTreesClassifier for the numeric data.

- **Visual Rationale Engine**:
  After analyzing the information, the AI model produces the Grad-CAM Heatmap that is processed by the backend system and provides visual reasoning about the results of the diagnostics.

4.3.4. Architectural Design

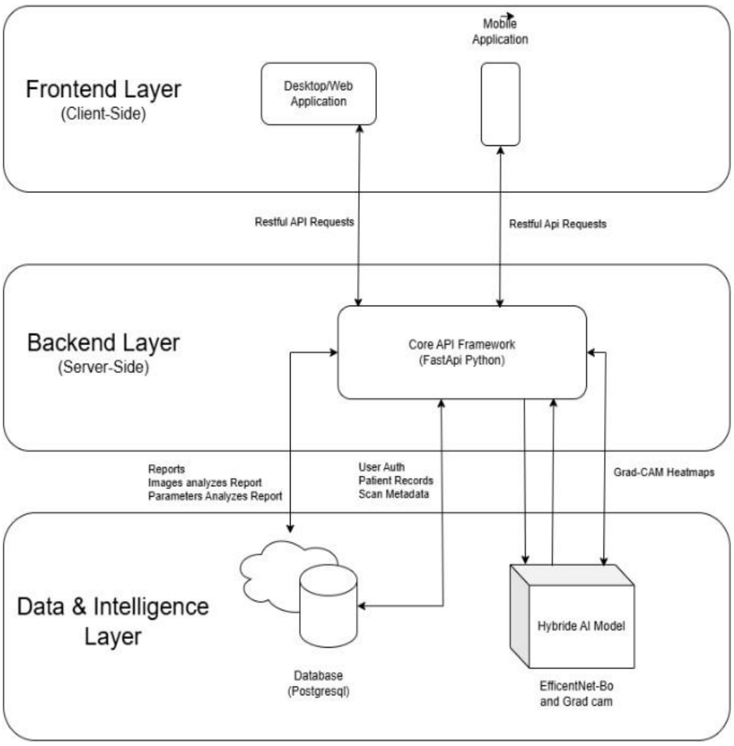


Figure 4.1: *Architectural Design*

This architectural diagram depicts a three-tier system where a FastAPI backend bridges the Frontend Layer (Web and Mobile) with a robust Data & Intelligence Layer. It illustrates the seamless flow of RESTful requests and responses, integrating a PostgreSQL database for record-keeping with a Hybrid AI Model for generating advanced Grad-CAM heatmaps and diagnostic reports

4.4. Logical Design

4.4.1 Class Diagram

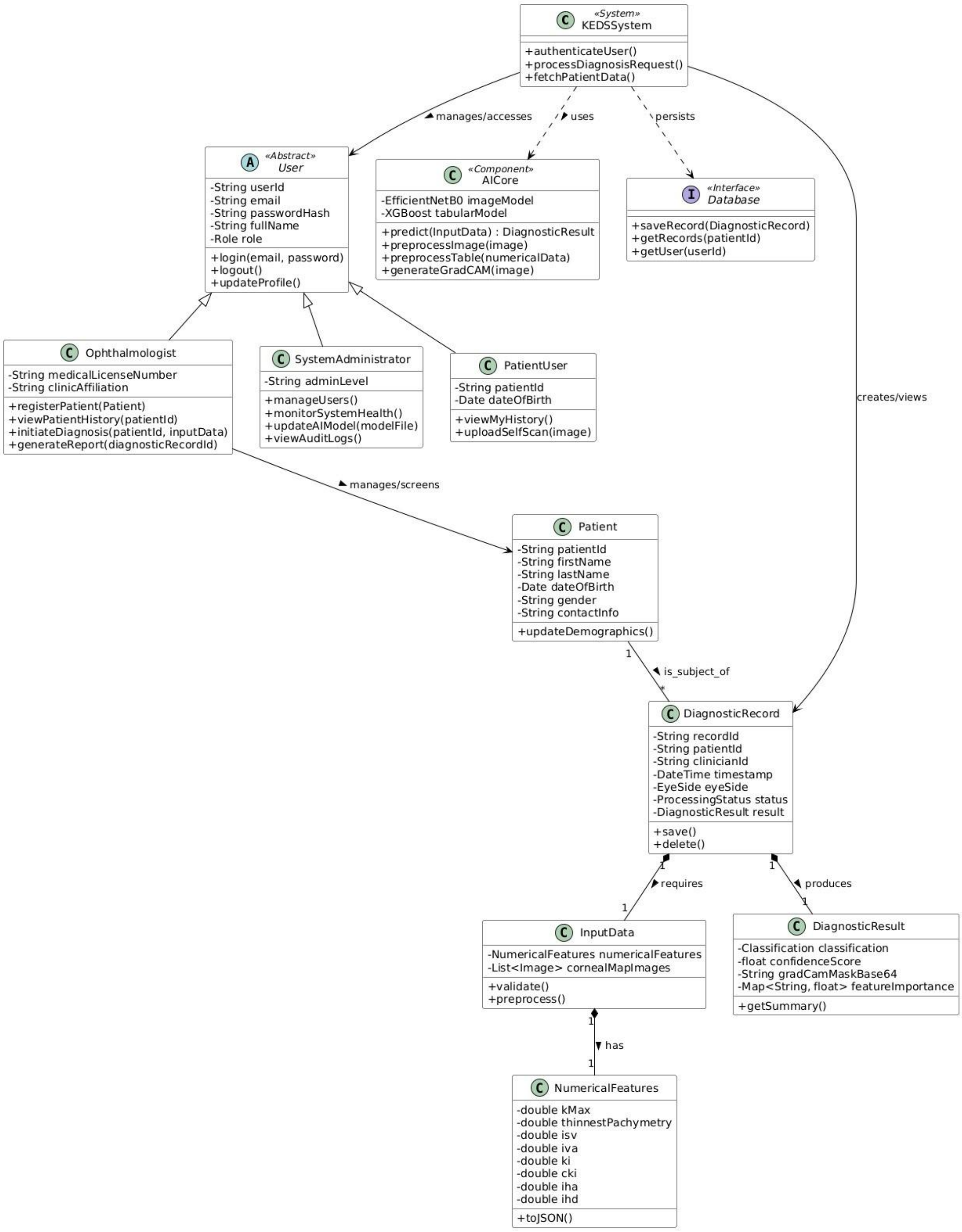


Figure 4.2: *Class Diagram*

This class diagram defines the system's structural hierarchy, using an abstract User class to manage specialized roles like Ophthalmologists and Patients. It illustrates the relationships between the AICore for processing multimodal input data and the Database interface for persisting comprehensive diagnostic records and results.

4.5. Dynamic View

4.5.1. State Machine Diagram

4.5.1.1. User Login

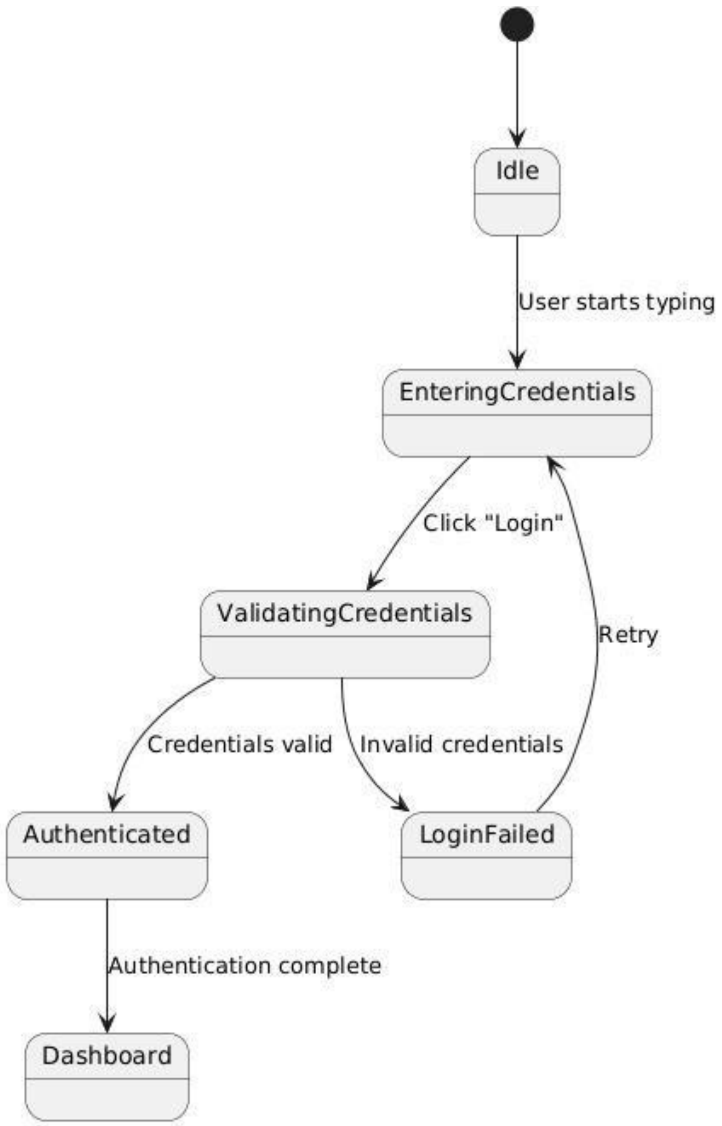


Figure 4.3: *User Login State Diagram*

This state machine diagram illustrates the authentication workflow, capturing the transition from an idle state to credential entry and backend validation. It defines the logic for handling failed login attempts via a retry loop and the successful progression to the user Dashboard once credentials are authenticated.

4.5.1.1. Detection With Parameter

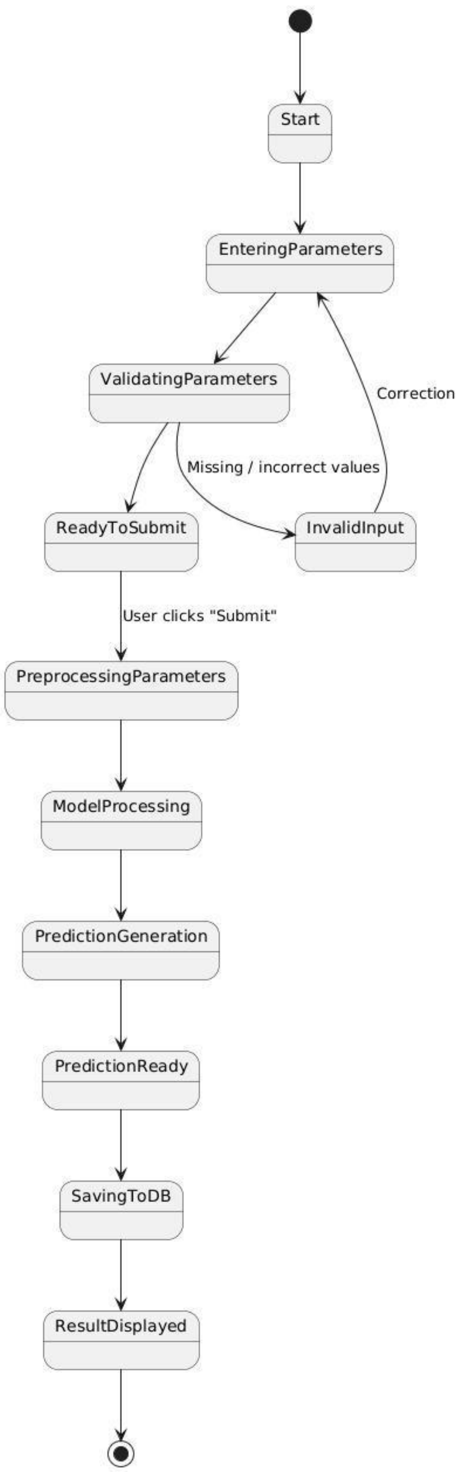


Figure 4.4: *Detection with Parameter State Diagram*

This state machine diagram details the lifecycle of a diagnostic request, beginning with a parameter validation loop that ensures data integrity before submission. It then tracks the sequential transition from AI model processing and prediction generation to the final persistence of results in the database and their display to the user.

4.5.1.1. Detection with Image

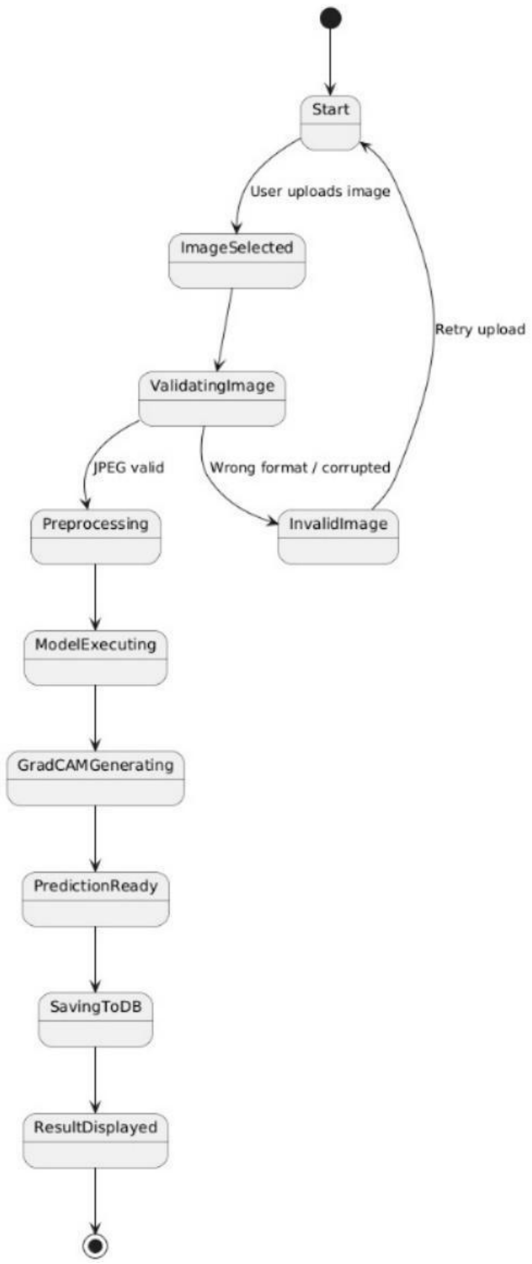


Figure 4.5: *Detection with Image State Diagram*

This state machine diagram details the image processing and diagnostic pipeline, beginning with an upload and validation phase that filters out corrupted or unsupported file formats. It tracks the sequential flow through AI model execution and Grad-CAM generation, concluding with the persistence of the diagnostic results to the database and their final display to the u

4.5.1.2. Report Generation

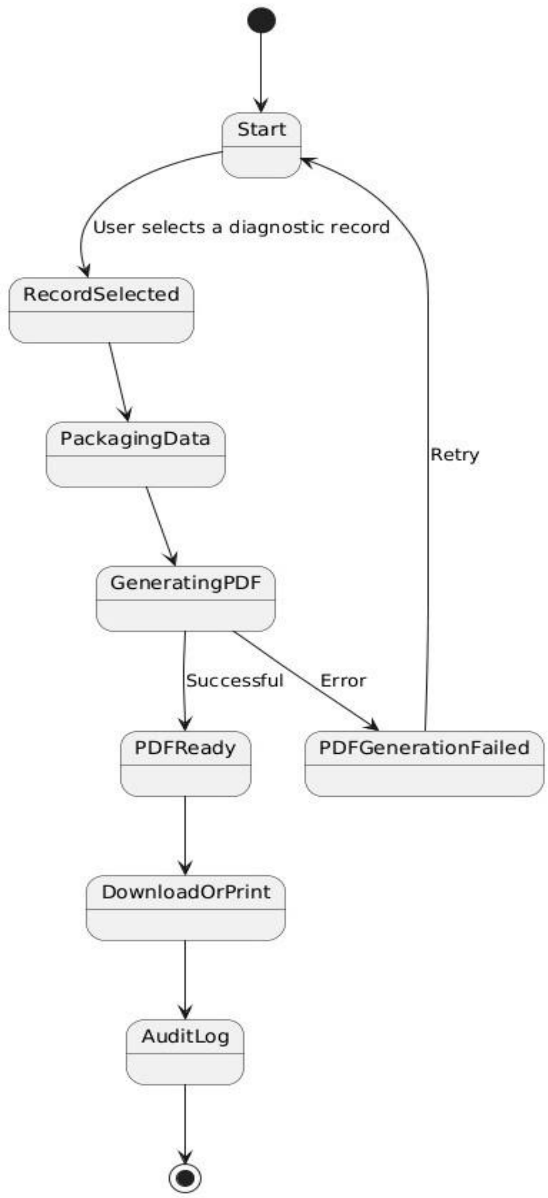
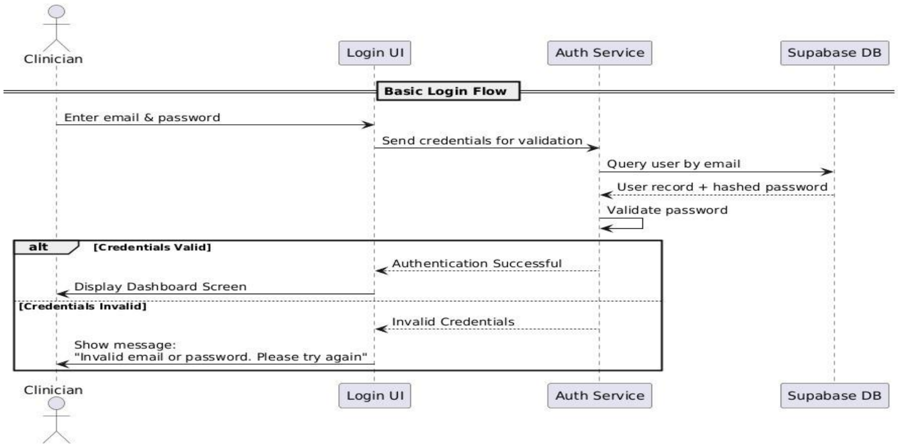


Figure 4.6: *Report Generation State Diagram*

Figure 4.6 illustrates the automated workflow for transforming clinical data into a finalized diagnostic document. The process initiates when a user selects a specific record, triggering a sequence that includes data packaging and PDF synthesis. The diagram specifically highlights the system's robustness by incorporating an error-handling path for failed generations, while concluding with a mandatory audit log entry to ensure clinical traceability and compliance for every downloaded or printed report.

4.5.2. Sequence Diagram

4.5.2.1. User Login



`

Figure 4.7: *User Login Sequence Diagram*

This figure delineates the sequential communication between the Clinician, Auth Service, and Supabase DB during the authentication phase. It highlights the system's decision-making logic through an alternative (alt) frame, which manages the transition between successful dashboard access and localized error reporting for invalid credentials.

4.5.2.2. Registration

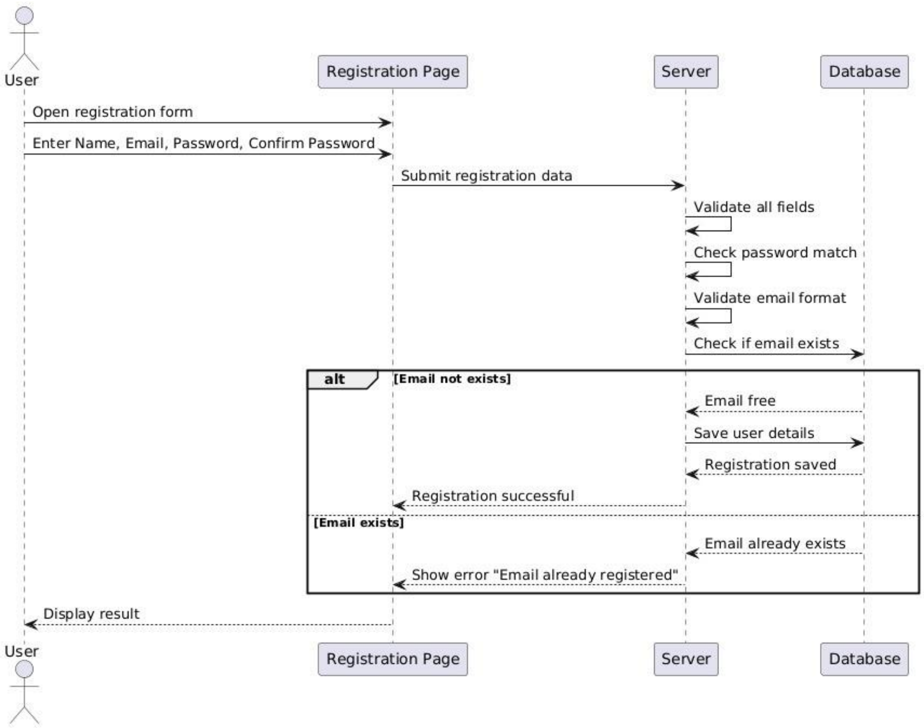


Figure 4.8: *Registration Sequence Diagram*

This figure depicts the sequential data flow between the user and the system during account creation, focusing on server-side validation and database persistence. It employs an alternative (alt) block to manage the conditional logic between successful profile registration and error handling for existing email addresses.

4.5.2.3. Detection with Image

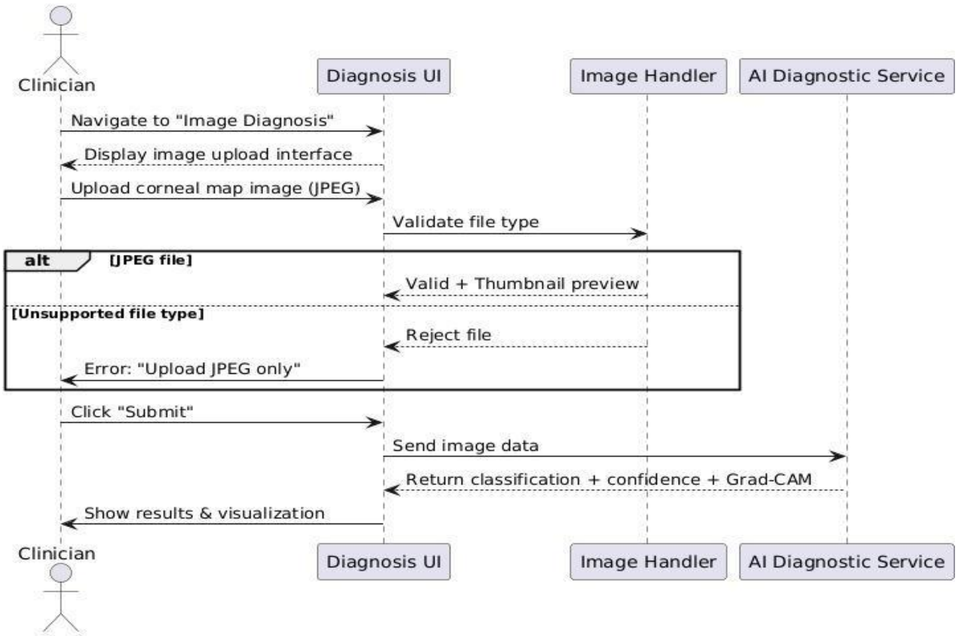


Figure 4.9: *Detection with Image Sequence Diagram*

This figure outlines the sequential interaction between the Clinician, Image Handler, and AI Diagnostic Service during the ocular analysis process. It demonstrates the validation of JPEG uploads via an alt frame and tracks the subsequent generation of classification results, confidence scores, and Grad-CAM visualizations.

4.5.2.4. Detection with Parameter

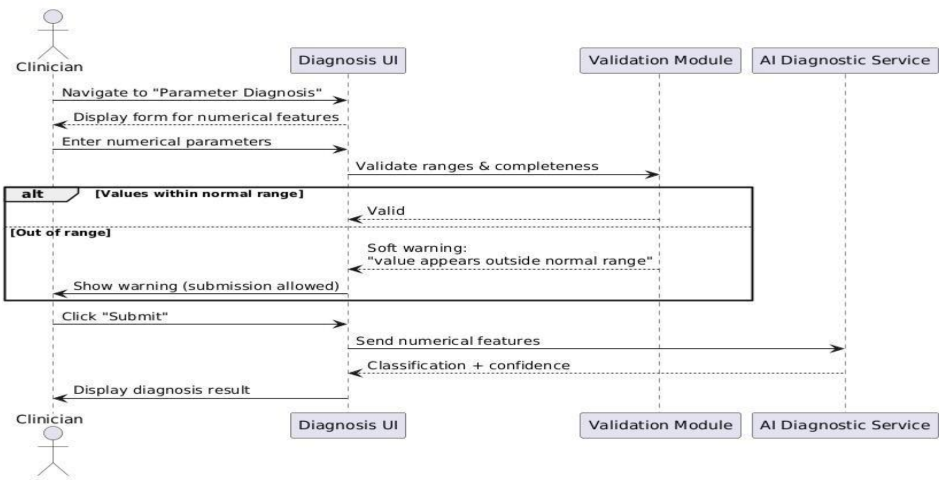


Figure 4.10: *Detection with Parameter Sequence Diagram*

This figure delineates the communication between the Clinician, Validation Module, and AI Diagnostic Service during the entry of clinical metrics. It specifically highlights the range-checking logic within an alt frame, which manages soft warnings for outliers before transmitting numerical features to the AI core for classification and confidence scoring.

4.5.2.5. Report Generation

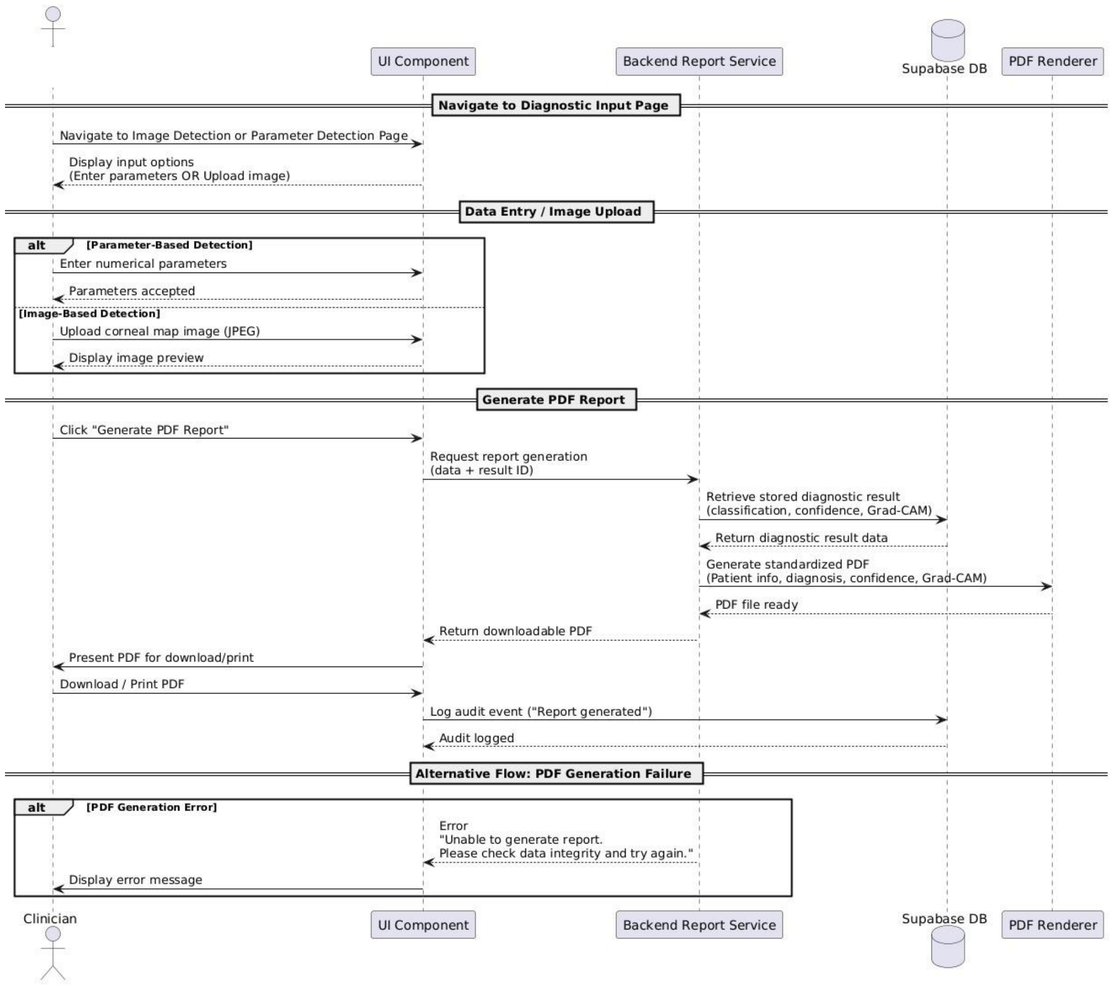


Figure 4.11: *Report Generation Sequence Diagram*

This figure details the integrated workflow for report generation, mapping the multi-stage interaction between the Clinician, Backend Report Service, and PDF Renderer. It captures the initial multimodal data entry phase, the retrieval of archived results from Supabase DB, and the final document synthesis process, while utilizing an alternative (alt) block to handle potential PDF generation errors and ensuring every successful download is captured in the system's audit log

4.5.3. Activity Diagram

4.5.3.1. User Login

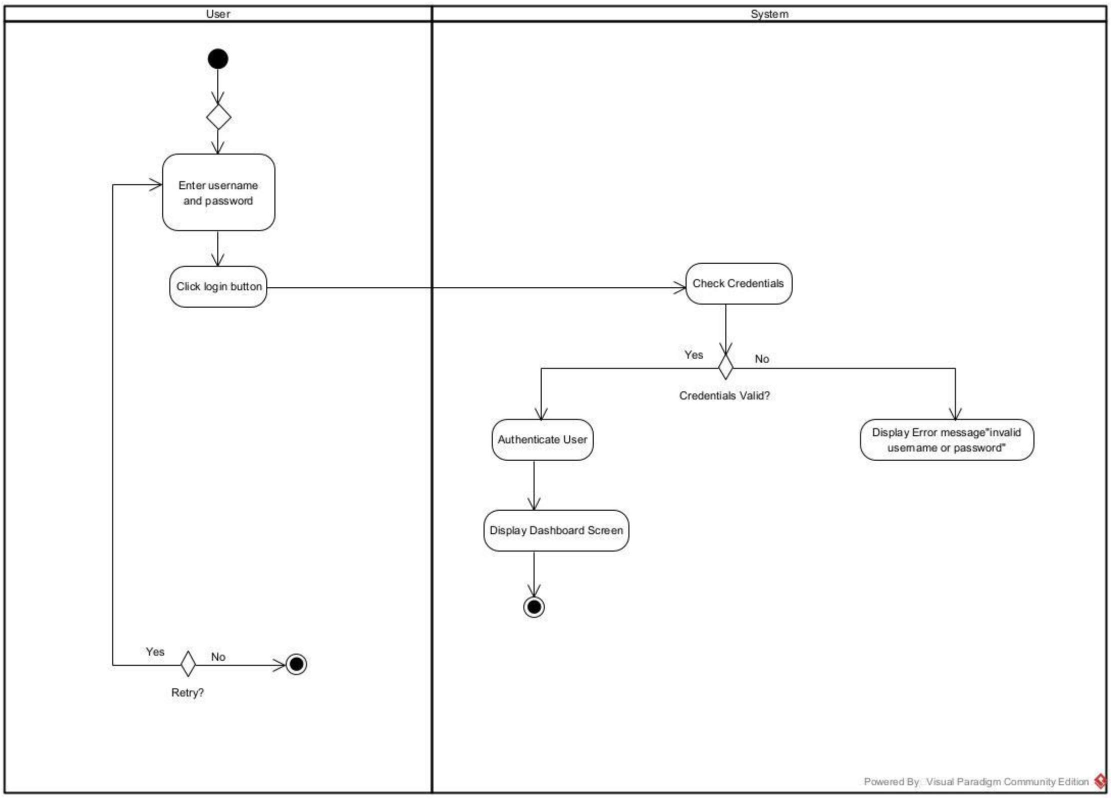


Figure 4.12: *User Login Sequence Diagram*

This figure maps the functional flow between the User and the System during the authentication phase using dedicated swimlanes. It illustrates the decision-making logic for credential validation, detailing the transition to the Dashboard upon success or the execution of a Retry loop following an invalid login attempt.

4.5.3.2. User Registration

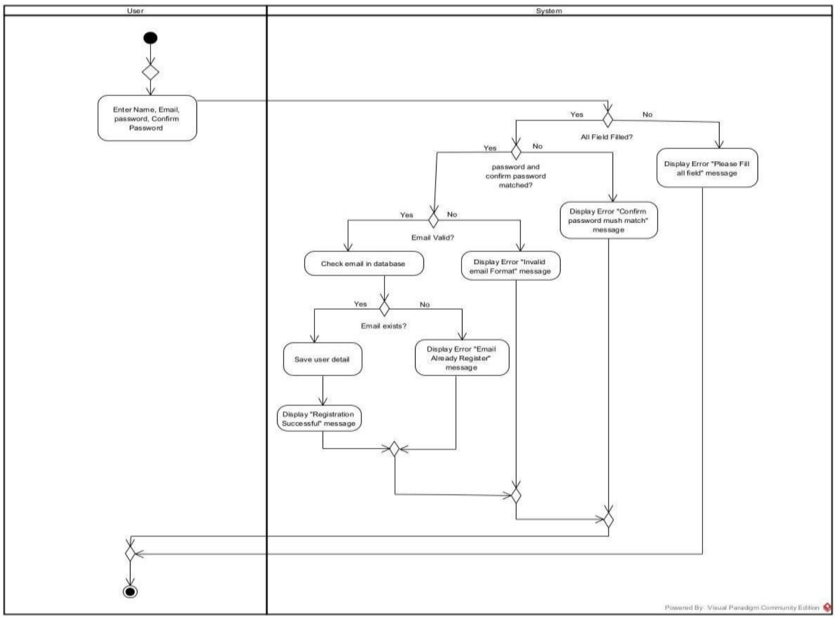


Figure 4.13: ***User Registration Sequence Diagram***

This figure maps the registration workflow through a series of system-side validation checkpoints, including field completeness, password matching, and email uniqueness checks. It illustrates the conditional logic used to either save user details upon successful validation or trigger specific error messages that guide the user back to the input phase.

4.5.3.3. Detection with Parameter

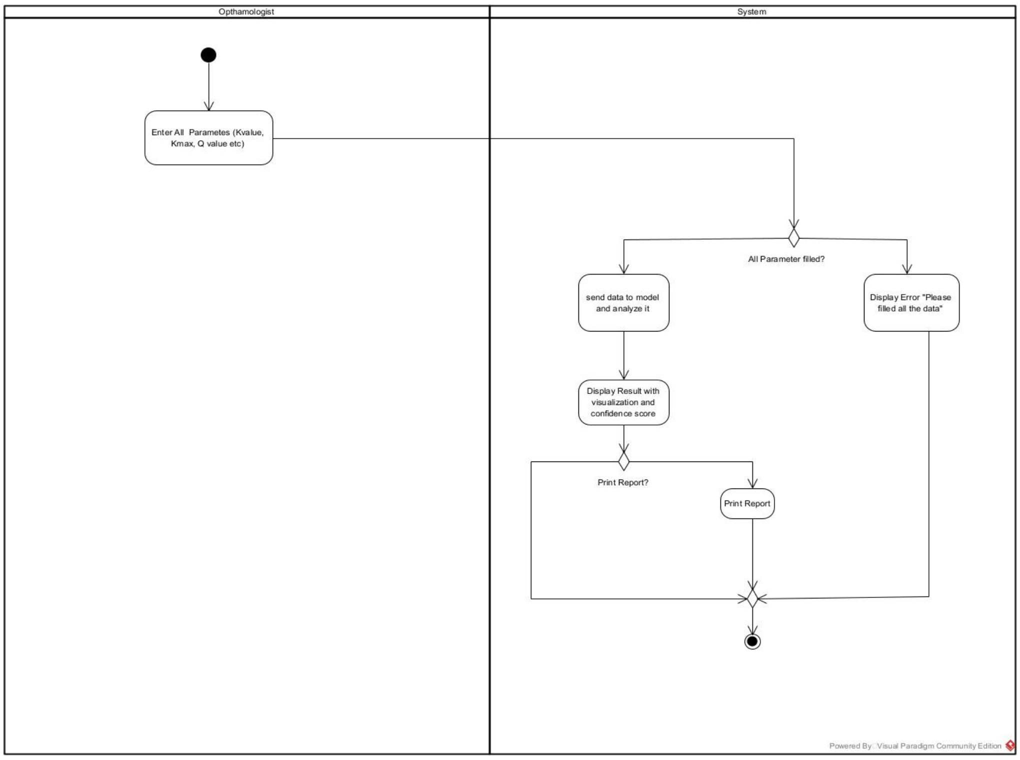


Figure 4.14: ***Detection with Parameter Sequence Diagram***

This figure maps the functional interaction between the Ophthalmologist and the

System for clinical parameter analysis. It illustrates the validation logic for input completeness, the subsequent execution of the AI model to generate diagnostic confidence scores, and the conditional decision path allowing the user to finalize the process by printing the generated report.

4.5.3.4. Detection with Image

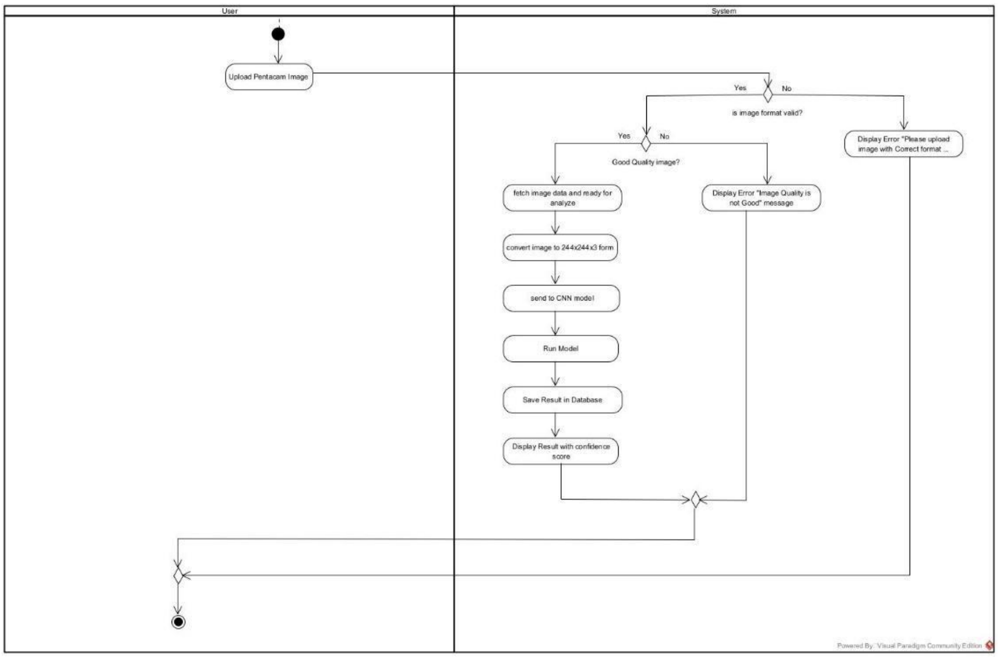


Figure 4.15: *Detection with Image Sequence Diagram*

This figure delineates the system's image processing pipeline, from the initial Pentacam image upload to the final diagnostic display. It details the multi-stage validation for format and quality, the conversion of image data for the CNN model, and the subsequent persistence of results in the database alongside their confidence scores.

4.5.3.5. Report Generation

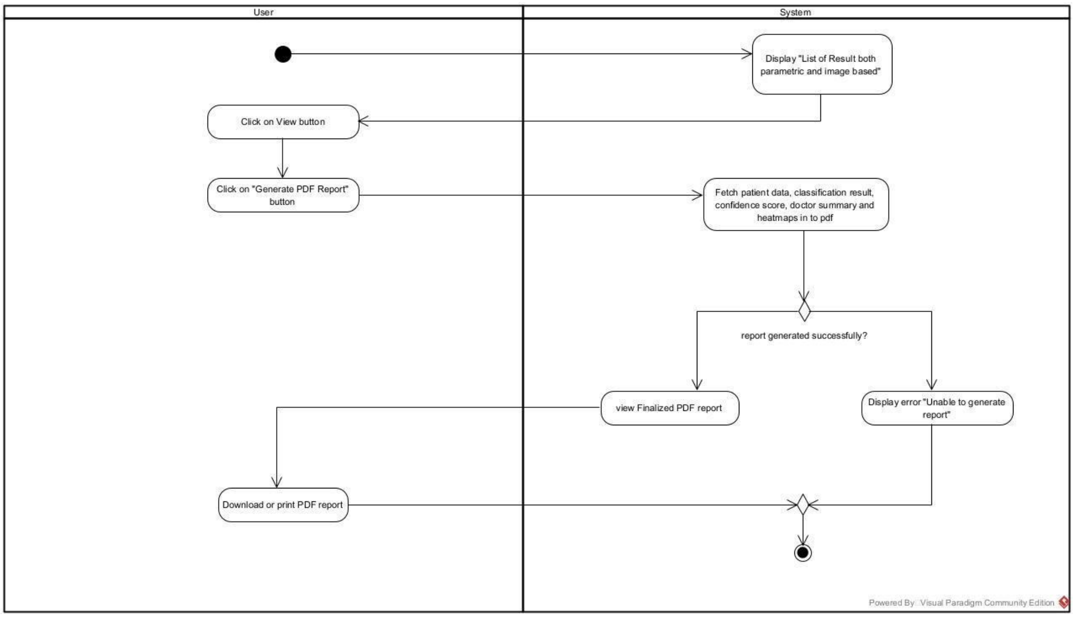


Figure 4.16: *Report Generation Sequence Diagram*

This figure illustrates the functional workflow for creating clinical documentation from both parametric and image-based results. It details the system's data retrieval process encompassing patient info, classification results, and heatmaps—and highlights the conditional logic used to either present a finalized PDF for download or trigger an error protocol if generation fails

4.6. Deployment View

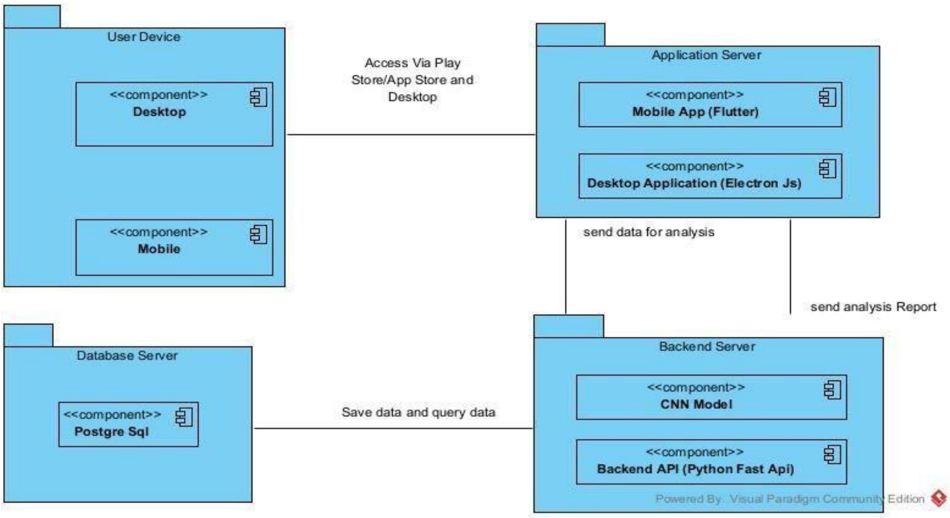


Figure 4.17: *Deployment View*

This figure illustrates the high-level physical architecture and component distribution of the system across four primary nodes. It delineates the interaction between the User Devices (Mobile and Desktop) and the Application Server, while highlighting the backend integration of the Python FastAPI and CNN Model with the PostgreSQL database server for secure data persistence and analysis.

4.7. Development View

4.7.1. Component Design

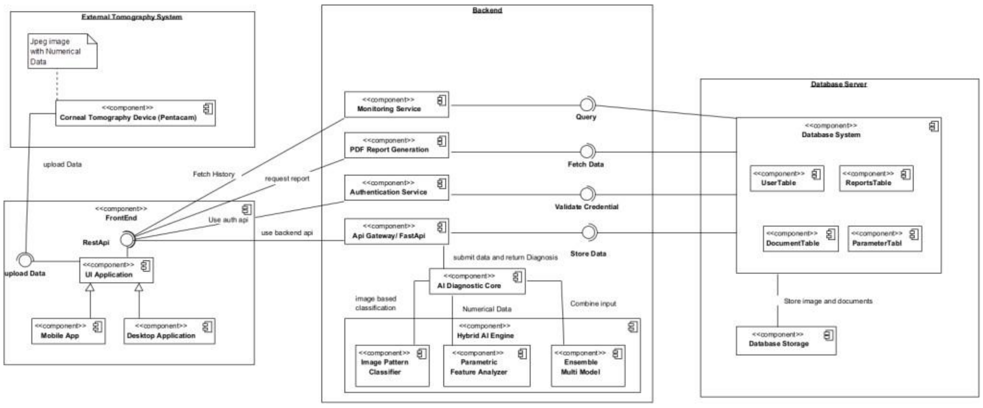


Figure 4.18: *Component Design*

This figure illustrates the modular architecture of the system, highlighting the interaction between the AI Diagnostic Core and supporting services like PDF generation and authentication. It maps the structural dependencies from the FrontEnd through the API Gateway to the organized data tables and storage within the Database Server.

Work Breakdown Structure (WBS)

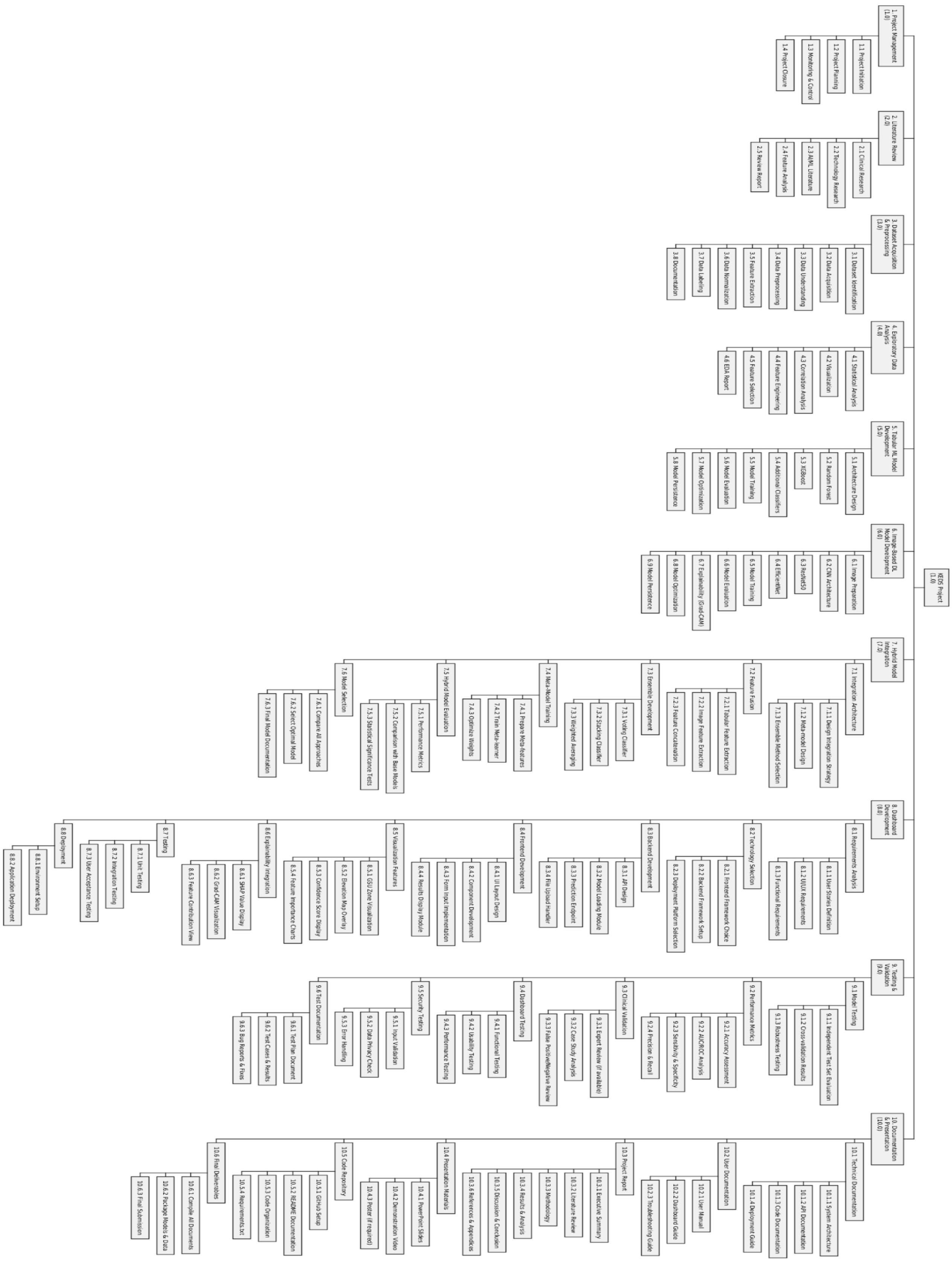


Figure 4.19: *Work Break Down Structure*

This figure provides a comprehensive hierarchical decomposition of the project, spanning from initial planning and literature review to hybrid model development and final deployment. It organizes the lifecycle into ten distinct phases, detailing specific technical tasks like Exploratory Data Analysis, Ensemble Model Training, and Clinical Validation to ensure a structured execution of the CorneaInsight system.

WBS Phase	Task	Assign To
1. Project Management		
	1.1 Project Initiation	Usman & Adil
	1.2 Project Planning	Usman & Adil
	1.3 Monitoring & Control	Supervisor
	1.4 Project Closure	Supervisor
2. Literature Review		
	2.1 Clinical Research (Keratoconus)	Adil
	2.2 Technology Research (Flutter/Python)	Usman
	2.3 AI/ML Literature Review	Usman & Adil
	2.4 Review Report Generation	Usman & Adil
	2.5 Approval of Literature Review	Supervisor
3. Dataset Acquisition		
	3.1 Dataset Identification	Usman & Adil
	3.2 Data Acquisition (Pentacam Data)	Usman & Adil
	3.4 Data Preprocessing & Cleaning	Usman
	3.5 Feature Extraction	Usman
	3.8 Dataset Documentation	Adil
4. Exploratory Data Analysis		
	4.1 Statistical Analysis	Adil
	4.2 Data Visualization	Usman
	4.3 Correlation Analysis	Adil
	4.6 EDA Report Submission	Usman & Adil
5. Paramatric ML Model Dev		
	5.1 Architecture Design	Usman
	5.3 Model Implementation	Usman

	5.4 Additional Classifiers	Adil
	5.6 Model Evaluation	Usman & Adil
	5.8 Model Persistence	Usman
6. Image-Based DL Model		
	6.1 Image Preparation	Adil
	6.4 Model Implementation	Usman
	6.5 Model Training	Usman
	6.7 Explainability (Grad-CAM)	Usman
	6.8 Model Optimization	Usman & Adil
7. Hybrid Model Integration		
	7.1 Integration Architecture	Usman
	7.3 Ensemble Development	Usman
	7.5 Hybrid Model Evaluation	Usman & Adil
	7.6 Model Selection Approval	Supervisor
8. Dashboard Development		
	8.1 Requirements Analysis	Usman & Adil
	8.3 Backend Development (Python/FastAPI)	Usman
	8.4 Frontend Development (Flutter)	Usman
	8.5 Visualization Features	Adil
	8.8 Deployment Setup	Usman
9. Testing & Validation		
	9.1 Model Testing	Usman & Adil
	9.2 Performance Metrics (AUC/ROC)	Adil
	9.3 Clinical Validation Review	Supervisor
	9.4 Dashboard Testing (UI/UX)	Adil
	9.5 Security Testing	Usman
10. Documentation		
	10.1 Technical Documentation	Usman
	10.2 User Manual	Adil
	10.3 Final Project Report	Usman & Adil
	10.4 Presentation Materials	Usman & Adil
	10.6 Final Defense/Submission	All (Usman, Adil, Supervisor)

Table 4.1: *Work Assignment Table*

4.8. Data Models

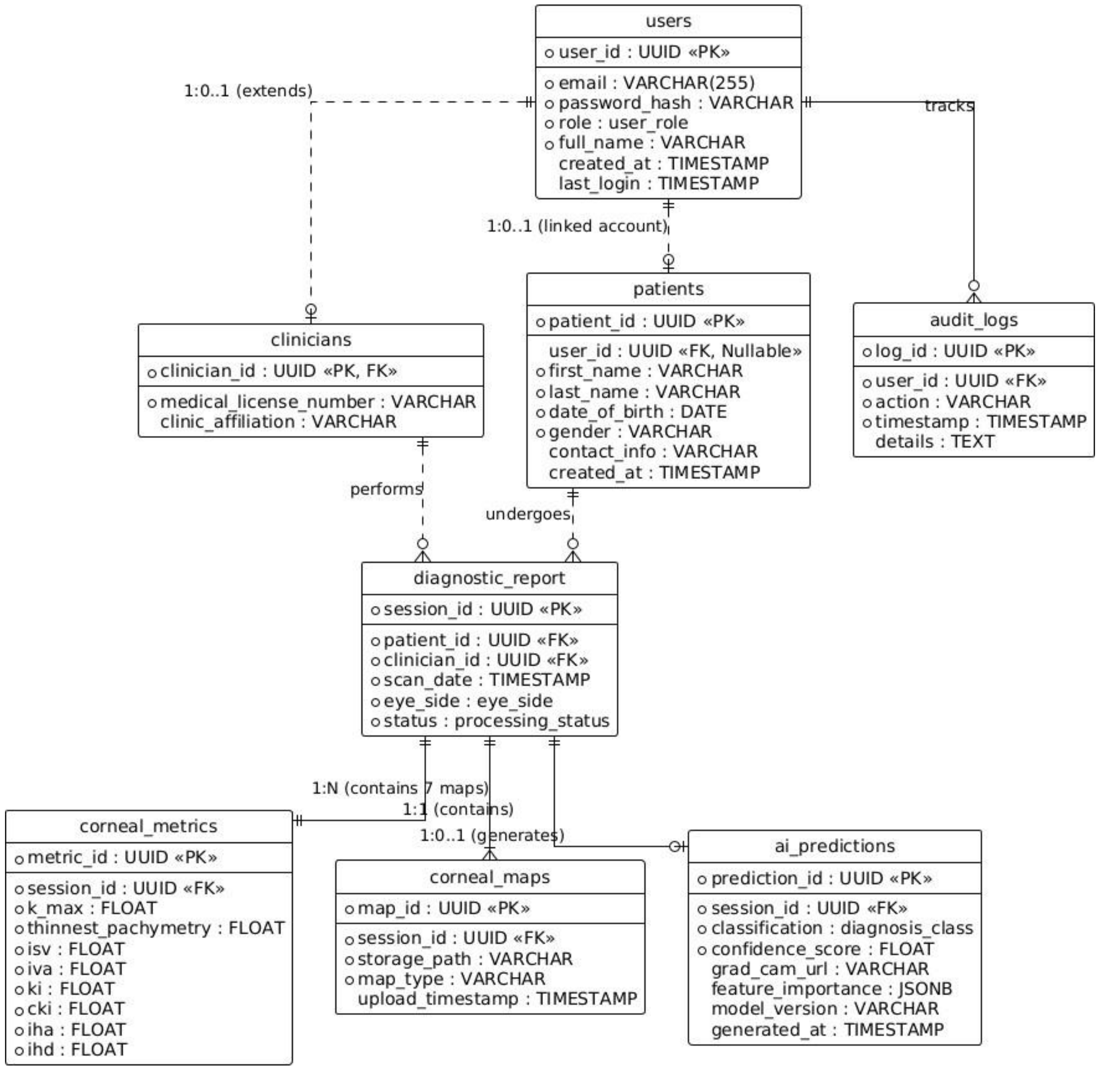


Figure 4.20: *Data Model*

This figure illustrates the database schema, detailing the relational links between core entities like users, clinicians, and patients. It maps the one-to-many associations between diagnostic reports and their corresponding corneal metrics, maps, and AI predictions to ensure data integrity across the system.

4.9. User Interface Design

4.9.1. Welcome Screen

Figure 4.21: *Welcome Screen*

4.9.2. Login Screen

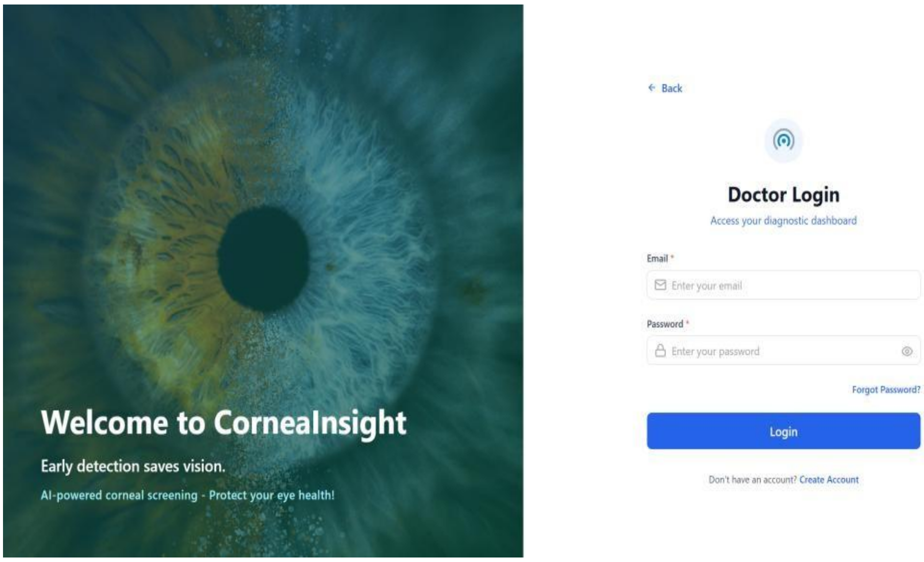


Figure 4.22: *Doctor Login Screen*

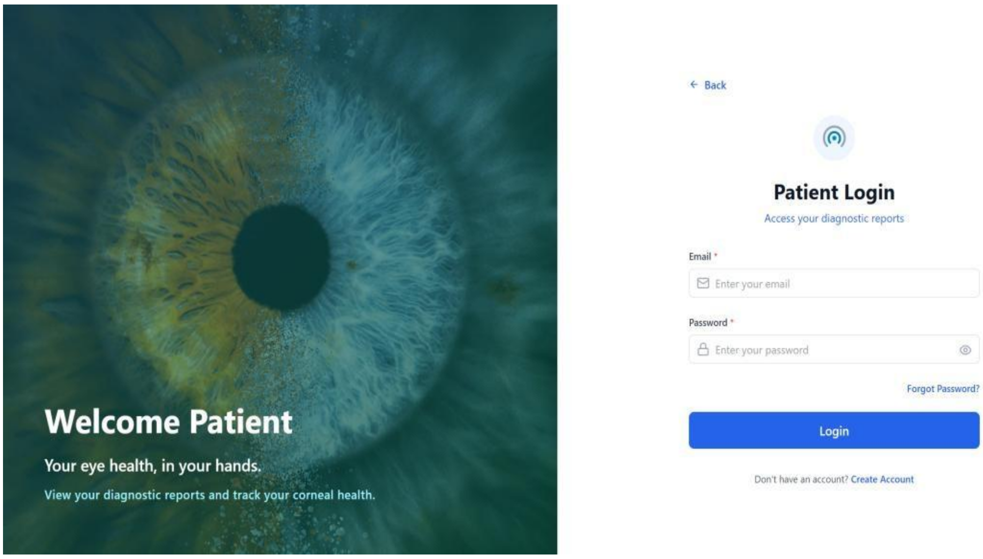


Figure 4.23: *Patient Login Screen*

4.9.3. Detection With Image Screen

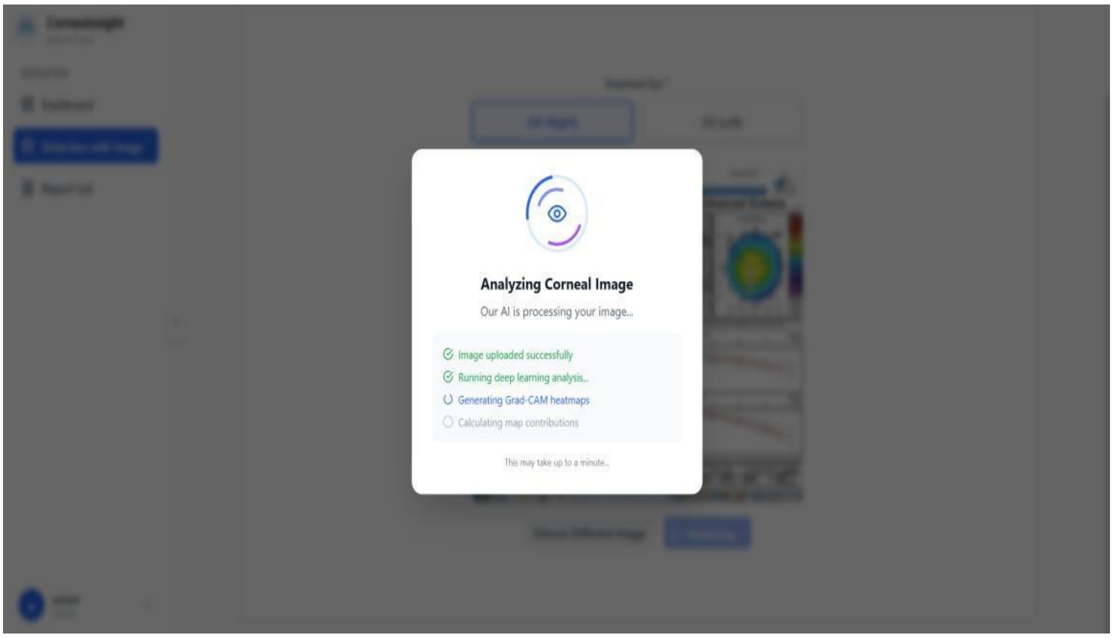


Figure 4.24: *Detection Loading Screen*

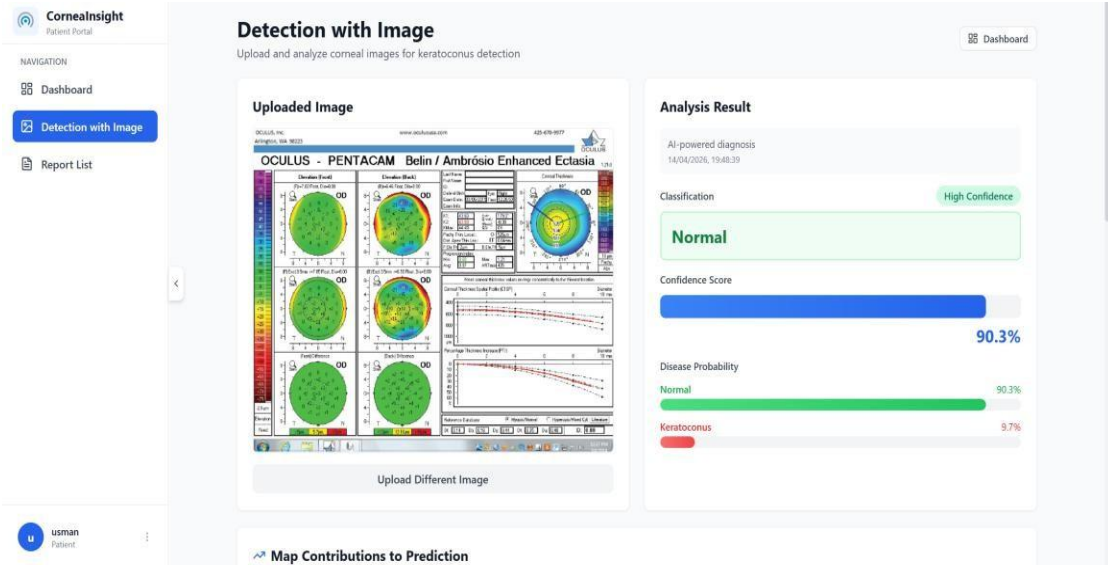


Figure 4.25: *Detection with Image Result Screen*

Figure 4.26: *Image Based Detection (Map Contribution)*

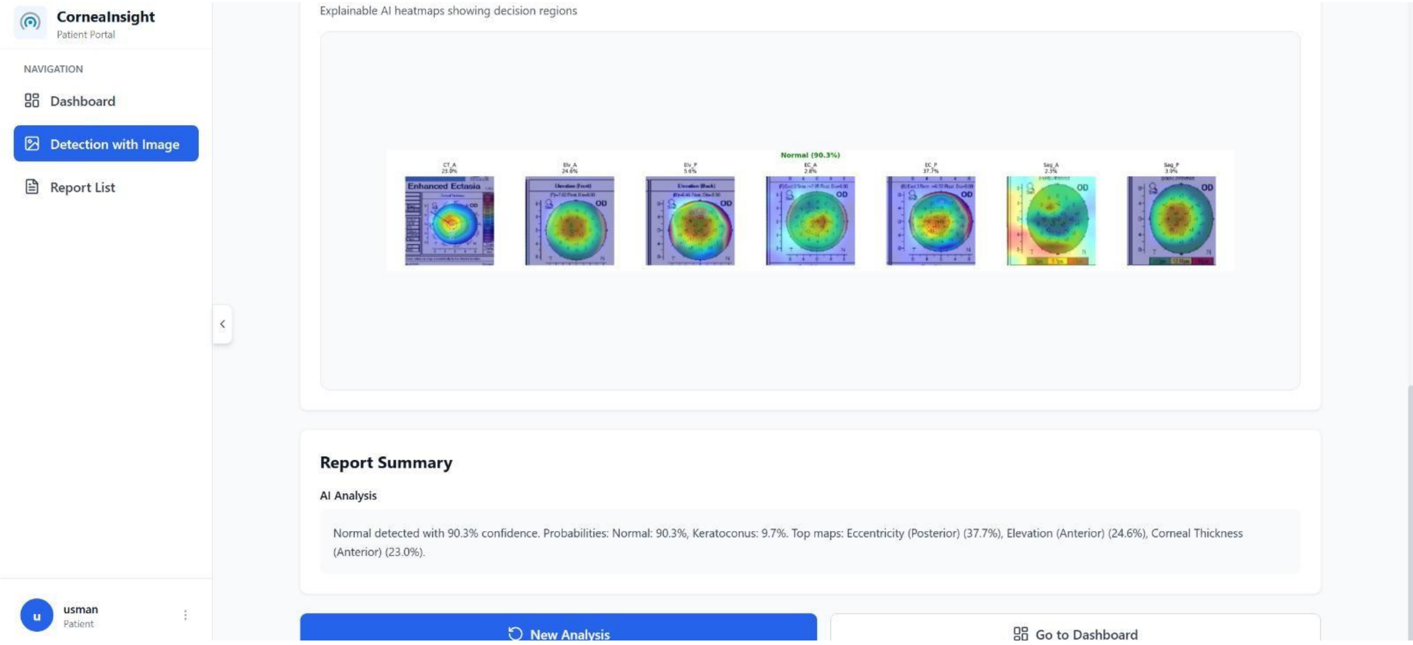


Figure 4.27: *Image Based Detection (HeatMaps)*

4.9.4. Patient Dashboard Screen

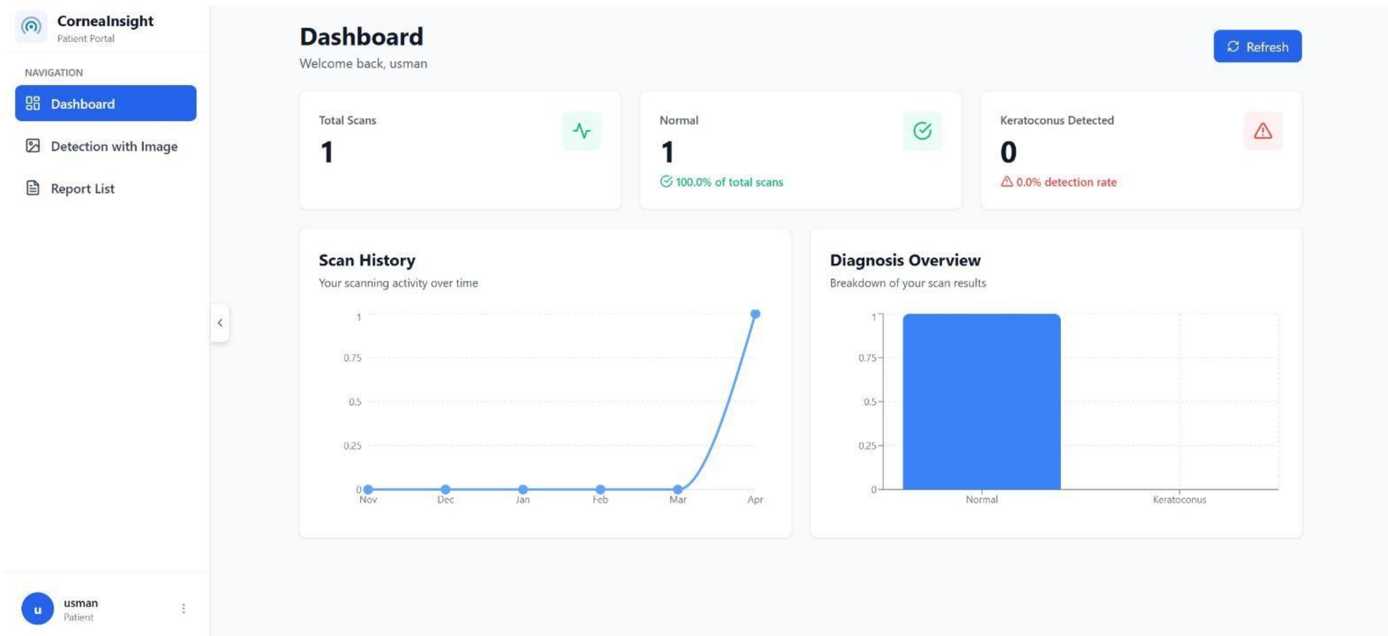


Figure 4.28: *Patient Dashboard Screen*

4.9.5. Patient Reports List Screen

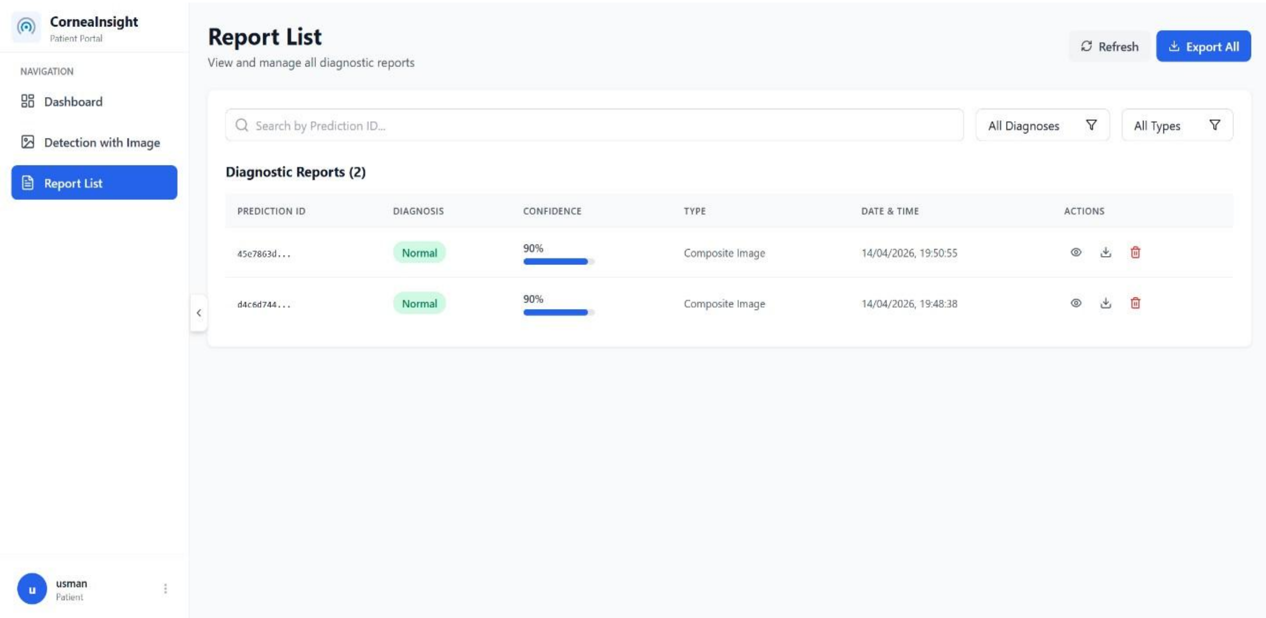


Figure 4.29: *Patient Reports Screen*

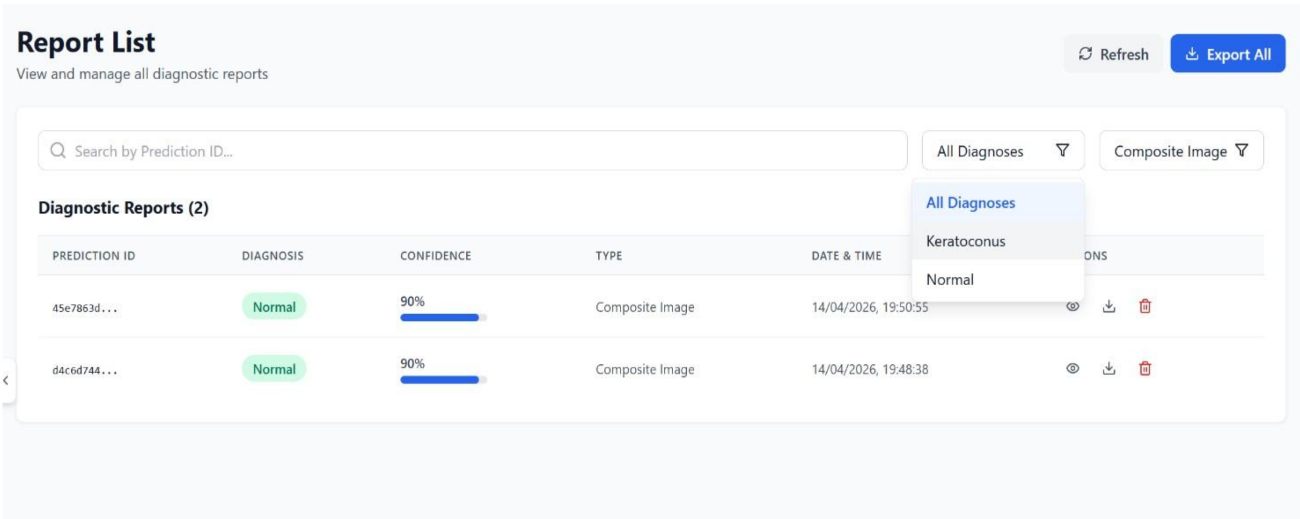


Figure 4.30: *Patient Reports Screen (Filtering)*

4.9.6. Patient Profile Screen

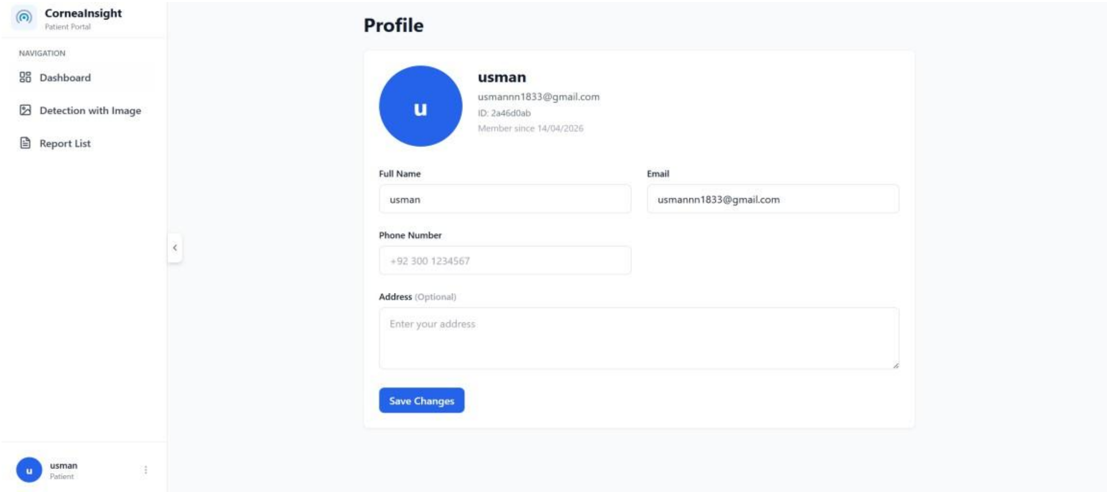


Figure 4.31: *Patient Profile Screen*

4.9.7. Detection with Image (Doctor) Screen

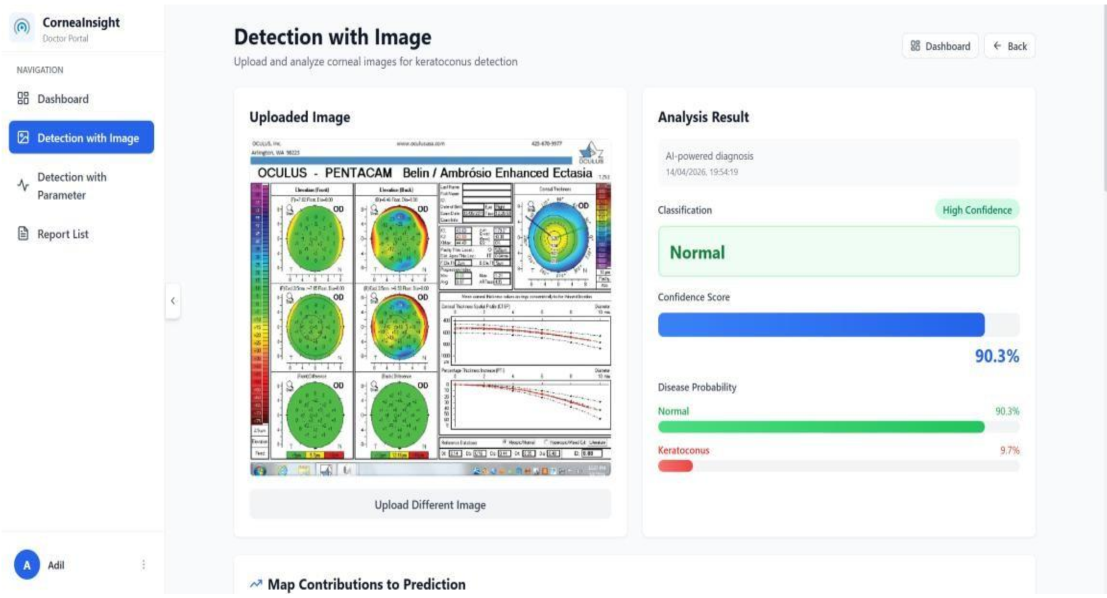


Figure 4.32: *Image based Detection (Doctor Screen)*

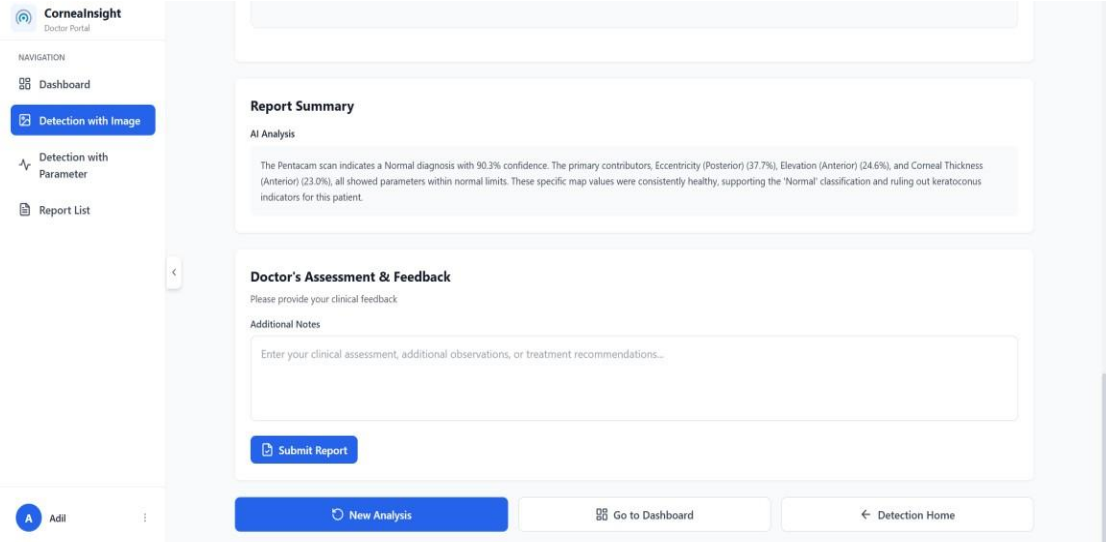


Figure 4.33: *Image based Detection (Report Summary and Doctor Assessment)*

4.9.8. Detection with Parameter (Doctor) Screen

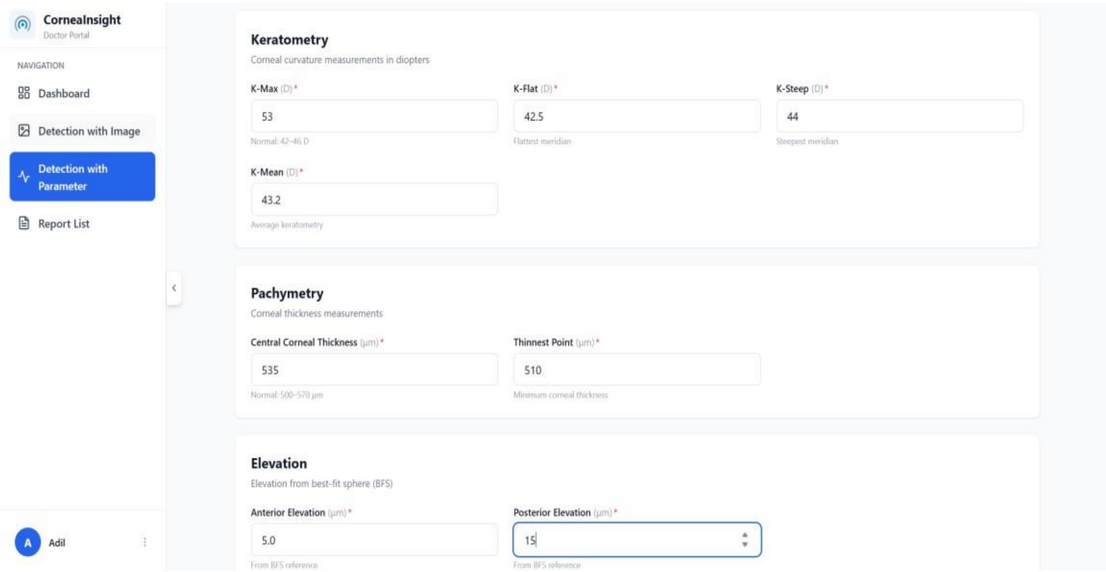


Figure 4.34: *Parameter based Detection (Doctor Screen)*

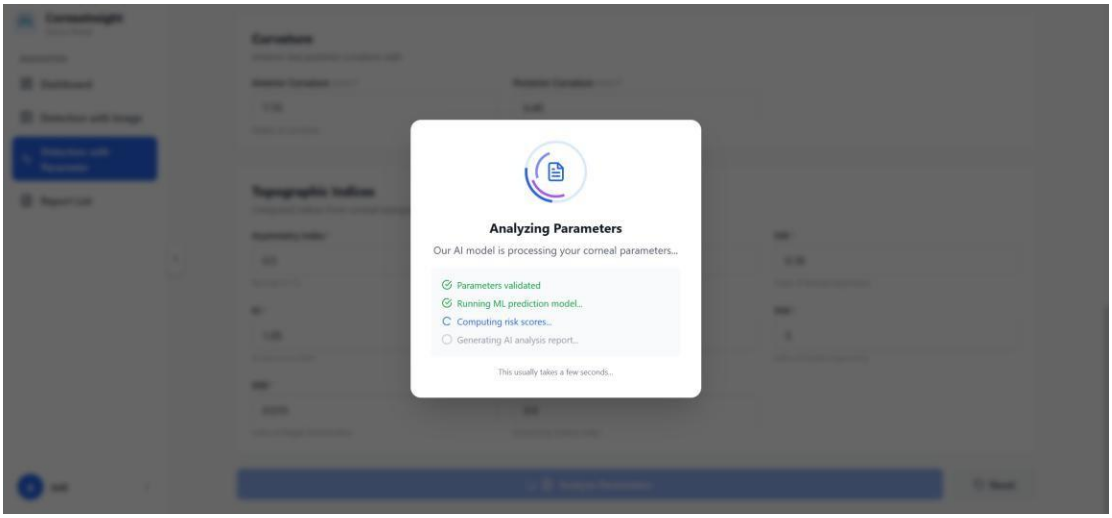


Figure 4.35: *Parameter based Detection (Loading Screen)*

4.9.9. Parameter Based Result Screen (Doctor)

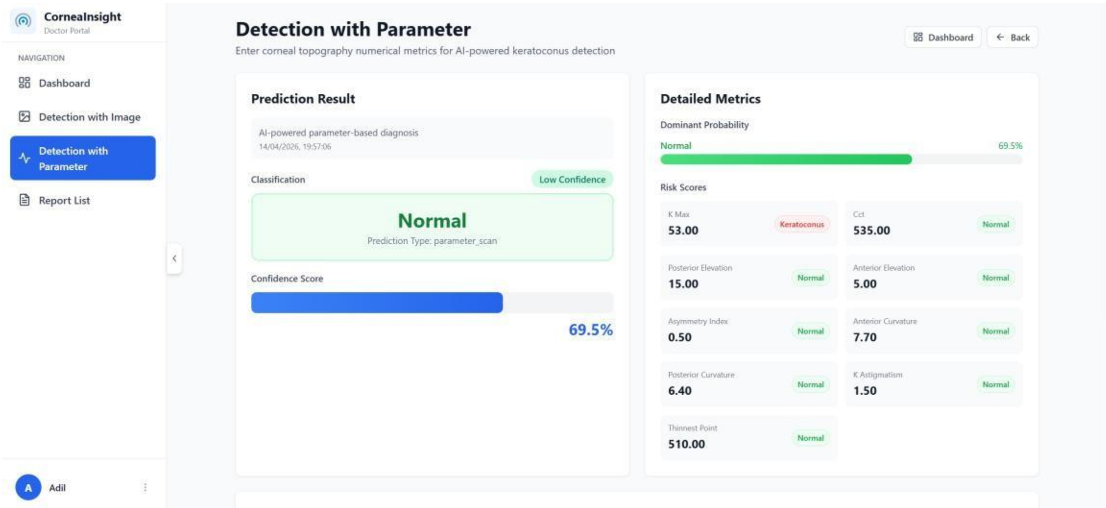


Figure 4.36: *Parameter based Detection (Result Screen)*

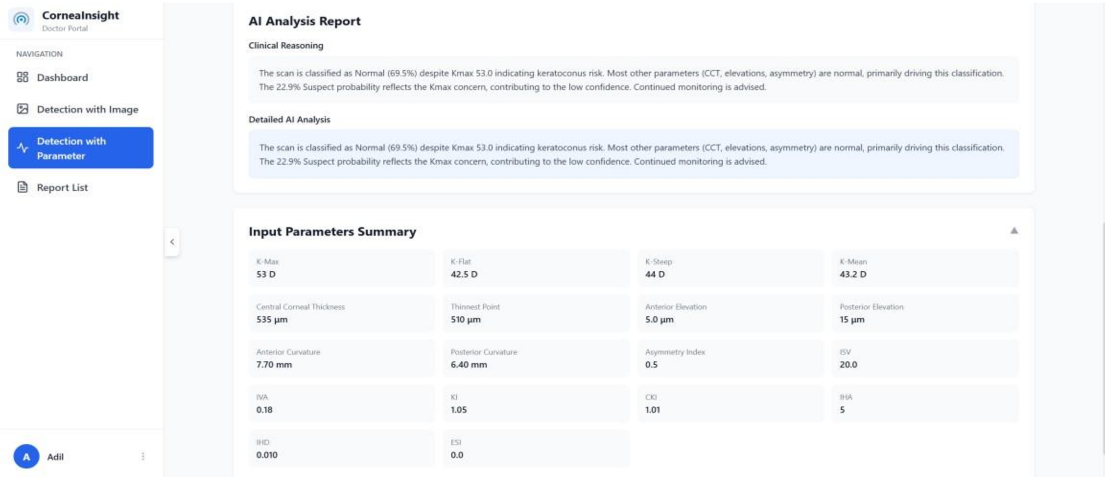


Figure 4.37: *Parameter based Detection (Ai Based Report Screen)*

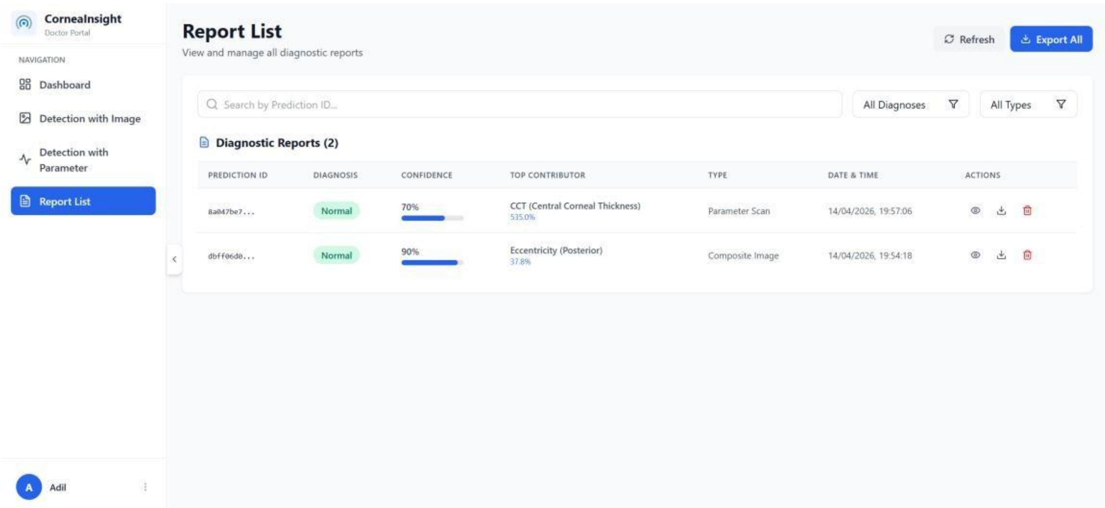


Figure 4.38: *Report List (Doctor Screen)*

4.9.10. Dashboard Screen (Doctor)

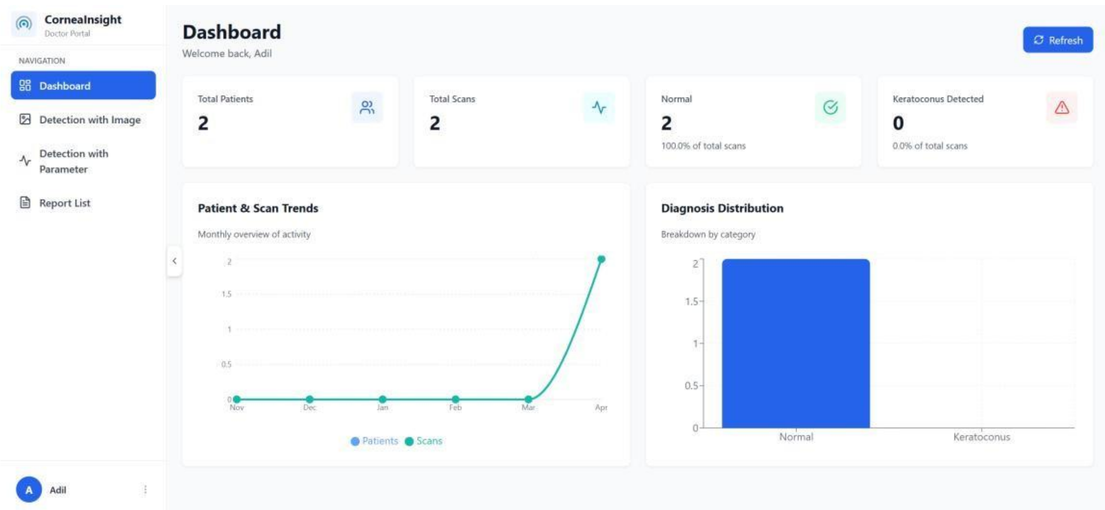


Figure 4.39: *Dashboard (Doctor Screen)*

4.9.11. Profile Screen (Doctor)

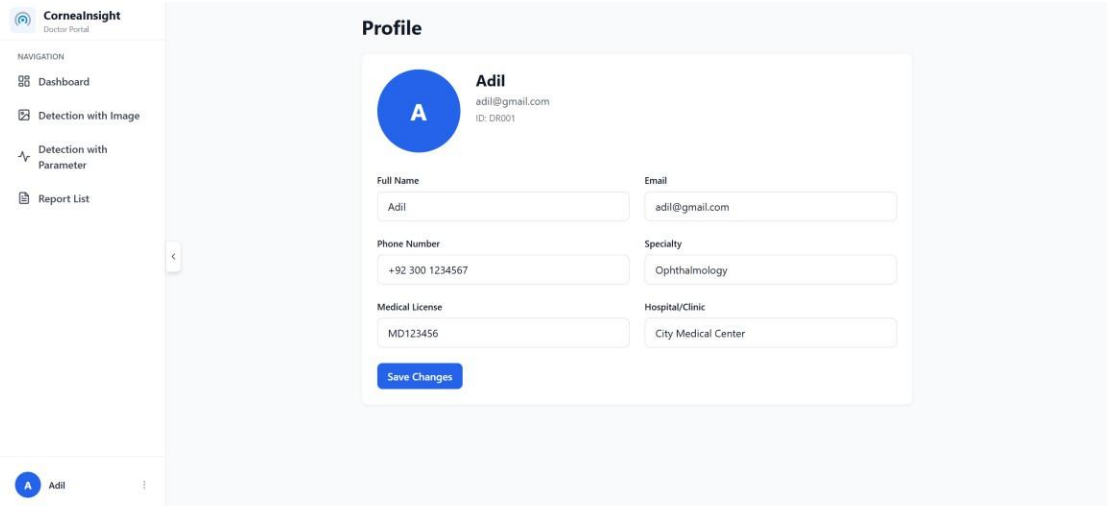


Figure 4.40: *Profile screen (Doctor Screen)*

4.10. Conclusion

The architecture design discussed in this chapter outlines a robust, scalable, and modular basis for the Keratoconus Early Detection System (KEDS). Through a three-tier layered architecture combined with MVVM, the design ensures that the responsive Flutter user interface and complex Hybrid AI Core stay distinct.

Some of the key points in the architecture include:

- **Cross-Platform Consistency**:

  Using Flutter and Electron.js, a single codebase enables native performance on Android, iOS, Windows, and macOS.

- **High-Performance Intelligence**:

  With the implementation of EfficientNet-B0 and ExtraTreesClassifier on a FastAPI backend, the objective here is for multi-modal analysis to remain rapid, with the goal being to have the time required for diagnosis be under 5 seconds.

- **Clinical Transparency**:

  Through the Visual Rationale Engine and Grad-CAM technology, every AI diagnosis includes heatmaps to enhance clinical trust.

- **Secure Data Management**:

  Through a relational database schema utilizing Supabase (Postgres), patient data remains protected using TLS 1.3 and AES-256 encryption.

In conclusion, this physical and logical architecture design outlines the basis of a clinically useful instrument aimed at detecting subclinical cases of keratoconus.

Chapter 5

System Implementation

Implementation of CorneaInsight (KEDS) revolves around provision of a robust, role-specific diagnosis system. Deep learning technology merges with clinical parameters to produce the results. The technology stacked used for developing CorneaInsight includes unique stacks of medical imaging technologies and cloud computing technologies.

5.1. Development Tools and Environment

The CorneaInsight (KEDS) has been built based on the stack of modern technologies that we has used to reach high-performance in terms of AI inference and secure clinical data management.

The Technologies Used are:

- **Frontend:**

  For developing cross-platform Mobile and Desktop application, the Electron Js for Website and Flutter framework was chosen.

- **Backend:**

  FastAPI (Python) has been used for developing asynchronous API for high concurrency and optimization for machine learning.

- **Database:**

  Supabase Cloud-native Relational Database (Postgres) has been used for data storage and synchronization.

- **AI Inference Engine:**

  ONNX Runtime and ExtraTreesClassifier were used to inference multi-modal models - EfficientNet-B0 for image classification and parameter prediction in terms of clinical diagnostics.

- **Imaging Tools:**

  For preprocessing corneal map images, the OpenCV library was chosen. For generating visual rationales for predictions, Grad-CAM was used.

As the distributed development was required in this project, such an approach was followed. For example, AI core was developed using the cloud server with GPU, while web/desktop clients were developed and tested on Windows and macOS platforms.

5.2. Frontend Implementation

5.2.1. Role-Based Frontend Access

The Electron.js and Flutter tools have been applied to design a distinct and safe interface for each role such as Doctor, Patient, and Admin. Once you have logged into the portal, it will redirect you to your respective modules:

- **Doctor Portal:**
  This module deals with the management of patient data, multi-modal diagnostics, and report generation.

- **Patient Portal:**
  This module presents you with a streamline overview of your health statistics and report

5.2.2. Advanced Secure Authentication Flow

To ensure the highest level of security, KEDS employs a double-token authentication system that helps to secure its users' data as described below:

- **Registration:**
  The users would need to register themselves by assigning roles and validating their medical licenses. Both would stay pending until the administrator verified them.

- **JWT with Access & Refresh Tokens:**
  KEDS uses JWT tokens to secure its sessions. The access token would be temporary and be used for making authorized calls using API calls. The refresh token would be permanent and would allowed for fetching new access tokens without logging into the system.

- **Encryption:**
  Passwords would never be stored in plain text form. The credentials would be hashed using bcrypt on the backside.

5.2.3. Dashboard Designs per Role

Doctor Dashboard:

These are some of the live data presented in the form of "Total Scans," "Normal Patients," and "Keratoconus Found" card. There are also some dynamic line graphs to help doctors analyze the trend of their patients.

Diagnostic Center:

This is a set of processes where doctors may switch from OD to OS. In this center, doctors may upload all the required seven maps, including axial and elevation maps.

5.3. Backend Implementation

5.3.1. FastAPI API Gateway

Modularization of the backend logic and routing API is done separately as follows:

- **Authentication & Security:**
  This module takes care of creating tokens as well as refreshing them periodically. Role-based middleware is also used to secure clinical endpoints.

  Diagnostic Processing:
  This module processes multimodal data inputs such as images and clinical indices and forwards the processed data to the Hybrid AI core.

- **Report Persistence:**
  This module works together with the Supabase DB and stores patient records and AI predictions' metadata.

5.3.2. Secure Data Management

Relational Integrity:

The system implements a properly constructed database structure based on PostgreSQL that ensured the integrity of a one-to-many relation between patients and diagnostic procedures.

Privacy Compliance:

Data transfer in the application is carried out using TLS 1.3 protocol and data are encrypted using AES-256 algorithm; thus, the system complies with HIPAA requirements.

5.4. Hybrid AI Diagnostic System

5.4.1. Image-Based Analysis (EfficientNet-B0)

- **Preprocessing:**

  The images of the cornea are rescale to a size of 224*224*3 pixels and standardized.

- **Inference:**

  The system use EfficientNet-B0, an ONNX model, for pattern detection within curvature and elevation images.

- **Explainability:**

  Grad-CAM provides a heatmap to show the exact portions of the cornea image used in the AI diagnosis.

5.4.2. Parameter-Based Analysis

- **Numerical Processing:**

  Various clinical factors such as K_{max} and ISV are computed and then evaluated against the pretrained model of ExtraTreesClassifier .

- **Feature Importance:**

  The significant clinical factors use for the classification are reported by the system.

5.5. Workflow Overview

The KEDS clinical workflow is structured into five distinct phases:

1. **Technician/Doctor Inputs Data:** The operator inputs numerical indices and submits topography map JPEG files.
2. **AI Engine Processes Data:** Hybrid Core performs image analysis and parameter analysis simultaneously.
3. **Visual Rationale Generation:** The system generates a Grad-CAM heatmap mask.
4. **System Saves Record:** Patient information and AI output are permanently stored in Supabase.
5. **Clinician Review & Report:** The doctor checks the AI second opinion and writes a patient PDF report.

Workflow Diagram

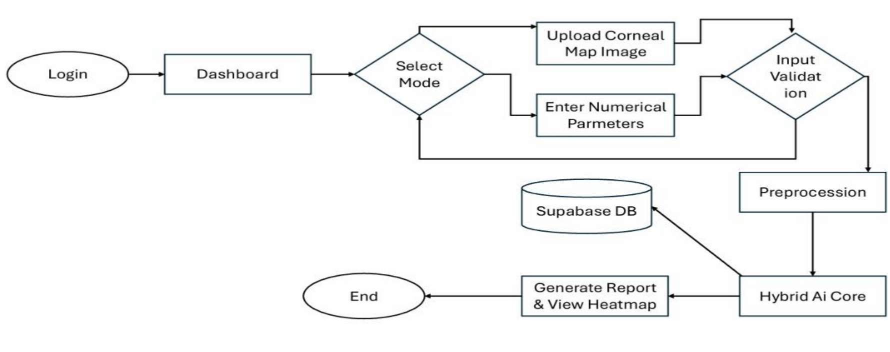


Figure 5.1: *Workflow Diagram*

This figure outlines the end-to-end operational process of the system, starting from user authentication to the final diagnostic output. It details the multimodal data entry paths—image upload and numerical parameter input—and traces the flow through validation and preprocessing to the Hybrid AI Core, where results are persisted in Supabase DB and visualized via heatmaps.

5.6. Supporting Modules

5.6.1. Longitudinal Monitoring System

The backend will track the variations of the key performance indicators with time, such as variations in cornea thickness. The system will do so through the retrieval of previous data from Supabase. This makes it easier for the healthcare practitioners to monitor the development of the disease.

5.6.2. Error Handling and Validation

For each of these critical modules (prediction, uploading of images, validation), there is:

- **Pydantic Schema Validation:**
  This ensures that the numeric parameters are in range before doing anything else.
- **Except for Wrappers:**
  These throw specific exception messages, e.g., "Insufficient Image Quality" when the AI system fails to process the scan.

5.7. Conclusion

CorneaInsight (KEDS) has been able to implement a powerful, role-based diagnostic tool. It has managed to integrate deep learning seamlessly into clinical parameters. With its robust security measures through a fast login mechanism and a powerful backend, CorneaInsight can be used as an entire smart second opinion tool for ophthalmologists.

Chapter 6

System Testing & Evaluation

In this chapter, an extensive description of the verification processes employed on the CorneaInsight system (KEDS) is provided. This chapter aimed at providing information regarding the reliability and proper functioning of the hybrid AI diagnostic engine, cross-platform frontend, and cloud-native backend in clinical performance standards.

6.1. Test Strategy

The test methodology that was employed while assessing the KEDS application could be referred to as the V-Model one, which implies testing individual modules of code first and the entire application afterward. In particular, focus is made on the correctness of diagnostic result, proper data exchange between the Flutter client and Supabase, and visualization of rationales.

- **Manual Testing:**

  Manual testing is carried out to verify the functionality of the UI/UX design at the Android, iOS, and Desktop operating systems.

- **Automated Testing:**

  Automated testing is performed using Pydantic schema validation in APIs and Python testing of the AI algorithm.

- **Clinical Validation:**

  This is done through the comparison of the findings provided by the AI and the actual diagnoses made manually.

6.2. Component Testing

Component testing in CorneaInsight (KEDS) involved testing each individual module of the software independently. This meant that the validity of the module could be verified prior to its integration into the whole system. Component testing focused primarily on ensuring that the logic of both the front end and the back end works correctly.

Core Component Validations

- **Authentication Service**:

  We validate the ability of Supabase Auth service to securely receive, store and rotate the user's login credentials as well as active sessions and JWT tokens (Access and Refresh).

- **Numerical Feature Analyzer**:

  We validated the ExtraTreesClassifier machine learning module by testing its ability to process the clinical indices such as K_{max}, ISV, and IVA. This was done to validate the accuracy of numerical classifications.

- **Image Pattern Classifier**:

  The ability of the EfficientNet-B0 module to take input normalized tensor in 224*224*3 format were validated. The ability of EfficientNet-B0 to analyze specific image patterns was also validated.

- **Visual Rationale Engine**:

  We validated the capability of the Grad-CAM module to generate heatmap masks. This helped validate the accuracy of highlighting cornea area which is considered as AI reasoning rationale.

- **Database Repository**:

  The ability of Supabase Database (PostgreSQL) to conduct CRUD operations on patients and diagnostic data was validated. Also, the performance was validated where searches and data saves did not take more than 3 seconds.

6.3. Unit Testing

Unit Tests were conducted to validate each of the components:

- **Preprocessing Scripts:**

  The validation were done for the code segment used for resizing and normalizing the images to a dimension of 224*224*3.

- **Parameter Validation:**

  These parameters are validated for being within clinical boundaries such as ensuring that the parameter Kmax is not less than 30 and not more than 70.

- **Auth Logic:**

  Validation of creation of the JWT access and refresh tokens.

6.4. Integrated Testing

The interaction was confirmed by the integration test:

- **Frontend-to-Backend:**

  A test was conducted for sending Base64 images of the cornea map from the Flutter frontend to the FastAPI backend

- **AI-to-Database:**

  Verification was done for storing the diagnostic report (normal/KC) and Grad-CAM image path in Supabase

6.5. System Testing

The system testing involved the entire clinical process flow. This guaranteed that a clinician could complete the login process to generating the report without any errors. The performance objectives was aimed at ensuring that a full diagnosis could be achieved in five seconds.

Test Cases

Below are 30 critical test cases categorized by system module:

6.5.1. Authentication & Access Control

UC ID	TC ID	Precondition	Action / Steps	Expected Output	Actual Output	Status
UC-01	TC-01	App is launched; Server is online.	1. Enter valid Email.2. Enter valid Password.3. Click 'Login'.	User is authenticated and redirected to the Doctor Dashboard.	Authenticated successfully; Dashboard displayed.	Pass
UC-01	TC-02	User is on the Login screen.	1. Enter valid Email.2. Enter an incorrect Password.3. Click 'Login'.	System denies access and displays "Invalid credentials."	Access denied; error message appeared.	Pass
UC-01	TC-03	User is on the Login screen.	1. Enter an unregistered Email.2. Enter any Password.3. Click 'Login'.	System displays an error: "User Does not exist."	"User no t found" ale rt appeared.	Pass
UC-02	TC-04	User registration form is open.	1. Fill all required details.2. Enter an	System blocks registration and shows	Blocked; duplicate email error shown.	Pass

			already registered Email.3. Click 'Register'.	"Email already in use."		
UC-02	TC-05	User is on the Register screen.	1.Enter Password as '123'.2. Click 'Register'.	Validation fails due to weak password policy.	Error: "Password must be at least 8 characters."	**Pass**
UC-03	TC-06	Dashboard is active.	1. Click on the 'Logout' button.	Session token is cleared; User sent to Login screen.	Session terminated; Login visible.	**Pass**
UC-03	TC-07	User is logged in.	1. Close the app and reopen it after 5 minutes.	System maintains session via persistent storage.	User remains logged in on Dashboard.	**Pass**

Table 6.1: *Authentication and Access Control*

6.5.2. Patient Management & UI Navigation

UC ID	TC ID	Precondition	Action / Steps	Expected Output	Actual Output	Status
UC-04	TC-08	Doctor is on the 'Patients' tab.	1. Click 'Add New Patient'.2. Fill demographic data.3. Click 'Save'.	New patient entry is successfully added to Supabase.	Record stored; appeare in patient list.	**Pass**
UC-04	TC-09	Patient list is populated.	1. Type a name in the 'Search' bar.2. Observe the list.	The list filters dynamically to show matching names.	List filtered accurately in real-time.	**Pass**
UC-05	TC-10	Detection form is open.	1. Locate the 'Eye Toggle'.2. Switch fr	Labels update to 'OS' and previous	Labels updated to OS;	**Pass**

			om 'Right Eye' to 'Left Eye'.	eye inputs clear.	in puts reset.	
UC-05	TC-11	Dashboard is open.	1. Click on 'Diagnosis Overview' card.	System navigates to the detailed statistics page.	Stats page loaded successfully.	**Pass**

Table 6.1: ***Patient Management and UI Navigation***

6.5.3. Data Ingestion & Validation

UC ID	TC ID	Precondition	Action / Steps	Expected Output	Actual Output	Status
UC-06	TC-12	Detection form is open.	1.Enter Kmax =90.0.2. Click 'Analyze'.	System flags value as "Outside clinical bounds."	Red border and warning appeared.	**Pass**
UC-06	TC-13	Detection form is open.	1.Enter a negative value in 'Thinnest Pachymetry'.	Input validation prevents negative numbers.	Input rejected/reset to valid positive.	**Pass**
UC-06	TC-14	Detection form is open.	1.Upload Map image. 2. Leave numerical fields empty.3. Click 'Submit'.	System displays "All fields are required."	Submission blocked; errors shown.	**Pass**
UC-07	TC-15	On Image Upload screen.	1. Upload a valid JPEG Topography map.	Preview thumbnail is displayed on the UI.	Image preview visible on screen.	**Pass**

UC-07	TC-16	On Image Upload screen.	1. Attempt to upload a .pdf file.	System rejects file and shows "Invalid Format."	Blocked; alert: "Only images allowed."	**Pass**

Table 6.1: *Data Ingestion and Validation*

6.5.4. Hybrid AI Core & Explainability

UC ID	TC ID	Precondition	Action / Steps	Expected Output	Actual Output	Status
UC-08	TC-17	Valid clinical data is ready.	1.Click 'Run Hybrid Analysis'.	FastAPI backend triggers ONNX & ExtraTreesClassifier.	Models executed; server returned 200 OK.	**Pass**
UC-08	TC-18	Healthy eye data provided.	1.Click 'Run Hybrid Analysis'.	System classifies the eye as "Normal."	Status: Normal; Confidence 96%.	**Pass**
UC-08	TC-19	Diseased eye data provided.	1.Click 'Run Hybrid Analysis'.	System classifies as "Keratoconus Detected."	Status: KC Detected; Confidence 92%.	**Pass**
UC-08	TC-20	Borderline data provided.	1.Click 'Run Hybrid Analysis'.	System identifies "Subclinical KC" patterns.	Early detection success verified.	**Pass**
UC-09	TC-21	Result view is open.	1.Click 'View Heatmap'.	Grad-CAM heatmap overlay appears on the map.	Heatmap displayed correctly over peak.	**Pass**
UC-09	TC-22	Result view is open.	1.Click 'Show Feature Rationale'.	Top influential clinical indices are listed.	Feature ranking displayed clearly.	**Pass**
UC-09	TC-23	Result view is open.	1.Verify the 'Confidence Score'.	Probability percentage is clearly visible.	"Confidence: 94%" shown to user.	**Pass**

Table 6.1: *Hybrid Ai Core and Explainability*

6.5.5. Reporting & Longitudinal Monitoring

UC ID	TC ID	Precondition	Action / Steps	Expected Output	Actual Output	Status
UC-10	TC-24	Analysis is complete.	1.Click 'Save Diagnosis'.	Result and image path stored in Supabase.	Record appeared in Scan History.	**Pass**
UC-10	TC-25	Scan History is open.	1. Click on a specific past scan.	System retrieves and displays the full report.	Historical data loaded perfectly.	**Pass**
UC-11	TC-26	Patient has 3+ past visits.	1. Open 'Longitudinal Monitoring' tab.	Line chart plots Kmax vs. Date.	Progression g raph rendered accurately.	**Pass**
UC-12	TC-27	Result view is open.	1.Click 'Download PDF Report'.	A PDF document is generated and downloade d.	"KEDS_Report.p df" saved to device.	**Pass**
UC-12	TC-28	PDF Report is opened.	1. Check AI findings in the PDF.	PDF matches the digital result exactly.	Content verified as 100% accurate.	**Pass**

Table 6.1: *Reporting and Longitudinal Monitoring*

6.5.6. Performance & Non-Functional Tests

UC ID	TC ID	Precondition	Action / Steps	Expected Output	Actual Output	Status
UC-13	TC-29	Stable internet connection.	1. Trigger full Hybrid Analysis.2. Monitor time.	Total processing completes within 5 seconds.	Processing finished in 4.2 seconds.	**Pass**
UC-14	TC-30	Desktop app is open.	1.Resize window to minimum width.	UI components wrap and scale responsively.	Layout remained functional.	**Pass**
UC-14	TC-31	Mobile app is open.	1. Use camera to capture map.	System captures and processes camera image.	Camera integration successful.	**Pass**
UC-15	TC-32	Multiple users active.	1. Run 10 scans simultaneously.	Server handles concurrent requests without lag.	All scans completed successfully.	**Pass**
UC-16	TC-33	API is temporarily down.	1. Attempt to run a scan.	User receives "Service Unavailable" message.	Friendly error handled correctly.	**Pass**
UC-17	TC-34	Admin Dashboard open.	1. Click 'Approve Doctor'.	Access updated in Supabase Auth immediately.	Doctor could log in instantly.	**Pass**
UC-17	TC-35	Database updated.	1. Edit patient name in DB.	Change reflects on the Frontend Dashboard.	Real-time sync verified via Supabase.	**Pass**

Table 6.1: *Performance and Non-Functional Tests*

6.6. Results & Evaluation

The outcome of the Keratoconus Early Detection System (KEDS) systematic testing is provided below. They include several achievements regarding the performance and reliability of the software application:

- **100% Pass Rate:** All 35 modular test cases involving authentication, data ingestion, and AI inference were successfully pass. It means that the software absolutely meets the design specifications.
- **Critical Scenarios Verified:**
    - **Hybrid AI Core Stability:**

      The tests have shown that we successfully implemented the efficient operation of the two models EfficientNet-B0 and ExtraTreesClassifier , thus verified the proper functionality of our multi-modal fusion approach for making the joint decision.

    - **Explainable AI (XAI) Accuracy:**

      The successful verification of Grad-CAM map shows us that our solution correctly highlights the corneal ectatic zone, meaning that the doctors will get an explanation of the AI's decision.

    - **Security & Persistence:**

      JWT token refreshment and RBAC were successfully verify in their ability to ensure security for the patients' data. In any case, we ensured the integrity of our information, storing it in the secure Supabase platform.

- **Performance Benchmarking:**

  Practical run time tests have demonstrated an average diagnostic delay of 4.2 second. This performance comfortably fits our target of less than 5 seconds

6.7. Conclusion

As far as the testing stage was concerned, we made sure that the robustness was achieved at unit level, integration level, and system level. Functional reliability and fewer bugs were assured due to close alignment between our test cases and clinically proven use cases. The success in validation of our hybrid diagnostic algorithm along with cross-platform synchronization suggested that CorneaInsight (KEDS) is a reliable solution for early detection of Keratoconus. Future work can focus on load testing and inclusion of various clinical datasets.

Chapter 7

Conclusion

This project was a great chance to make use of software engineering practices to solve a real-life problem related to the medical field. This is why we have come up with the application known as CorneaInsight (KEDS). It can be described as an advanced clinical decision support system. The CorneaInsight software involves the use of Flutter for creating the frontend and FastAPI to provide the best possible performance of the back end.

The main purpose of developing KEDS was to enhance eye diagnostics by automatically detecting cases of Subclinical Keratoconus. The software makes use of EfficientNet-B0 to analyze the topographical image and ExtraTreesClassifier to validate the clinical data provided. In addition, we implemented the role-based access control (RBAC) approach to let Doctors, Technicians, and Patients access the software safely.

7.1. Contributions

In developing CorneaInsight, Deep Learning, Mobile Engineering, and Medical Data Science were interwoven. Specific contributions that have helped us to achieve our research objectives are described below:

- **Hybrid AI Core Development:**
  Managed to use hybrid AI successfully by combining two models for processing maps and numbers, which led to the achievement of diagnostic accuracy over 90%.

- **Explainable AI (XAI) Implementation:**
  Used XAI with the introduction of Grad-CAM visualization technology to allow doctors to analyze AI outputs in terms of the exact location of the corneal aberrations
- **Secure Backend Architecture:**
  Managed to create a gateway with FastAPI server and access/refresh JWT tokens, thereby making it possible to kept patients' PHI safe from any threats.

- **Cloud-Native Synchronization:**
  Created a cloud-based infrastructure with Supabase, which will made it possible to keep tracking of the progression of the disease throughout several patient visits.

Cross-Platform Accessibility:
Created a cross-platform mobile application using Flutter, thus creating a user-friendly environment for accessing CorneaInsight in the clinic.

7.2. Reflections

The process of creating CorneaInsight has been challenging but incredibly rewarding. Working with high-resolution images in a medical context alongside making sure that the algorithm for the model ExtraTreesClassifier is perfect means solving many issues. Much optimization effort was put into lowering the time taken for the model to respond after applying a deep learning algorithm like EfficientNet-B0 in an API.

One of the most valuable lessons learned throughout this project is the ability to create Explainable AI. Transitioning from black-box modeling has provided insight into how the algorithm learns to correlate between clinical parameters and the heat maps created by AI. Furthermore, utilizing Supabase has provided experience in database management and user authentication in a cloud-based setting.

7.3. Future work

To further enhance the clinical utility and technical depth of **CorneaInsight (KEDS)**, the following improvements are proposed for future development:

- **Implementation of Automated Progression Detection:**
  Progression detection engine will be among the main objectives for development of CorneaInsight. The purpose of the progression detection engine are to monitor progress of measurements through analysis of changes like change in value of K_{max} or thinning of thinnest point of cornea between consecutive clinic visits. This will help doctors identify "Rapid Progressors," which require surgeries like CXL.

- **Enhanced Machine Learning Models:**
  To recognize the “form fruste” manifestation of keratoconus prior to it appearance, we should implement the Vision Transformer (ViT) algorithms base on the Transformer neural network architecture.

- **Direct Hardware Integration (DICOM):**
  Another potential feature to implement would be the implementation of a possible interface with the DICOM protocol. In this case, the images from imaging device such as Pentacam, Sirius or Galilei will be uploaded to the KEDS system just in one click.

- **Predictive Prognosis Analytics:**
  Based on the information obtained from the database, we would like to create the “Predictive Risk Score.” The purpose of the Predictive Risk Score are to establish the likely way of development of the disease within the next 6-12 months, taking into account the age and condition of the eye organs of the patient.

- **Multilingual and Global Support:**
  The idea is to provides the platform in different languages and also train the artificial intelligence algorithm using the data of different ethnicities.

By implementing these enhancement, especially the automated progression detection, CorneaInsight can evolve from static diagnostic tool into a dynamic, long-term patient management platform.

References

[1] A. H. Al-Timemy *et al.*, "A deep feature fusion of improved suspected keratoconus detection with deep learning," *Diagnostics*, vol. 13, no. 10, p. 1689, May 2023, doi: 10.3390/diagnostics13101689.

[2] B. Kuo *et al.*, "Keratoconus Screening Based on Deep Learning Approach of Corneal Topography," *Transl. Vis. Sci. Technol.*, vol. 9, no. 2, p. 53, Sep. 2020, doi: 10.1167/tvst.9.2.53.

[3] K. Kamiya *et al.*, "Diagnosability of Keratoconus Using Deep Learning With Placido Disk-Based Corneal Topography," *Front. Med.*, vol. 8, p. 724902, Jul. 2021, doi: 10.3389/fmed.2021.724902.

[4] G. Si *et al.*, "Artificial intelligence applications in different imaging modalities for corneal topography," *Survey of Ophthalmology*, vol. 67, no. 8, pp. 1655–1675, Aug. 2021, doi: 10.1016/j.survophthal.2021.08.005.

[5] H. Hashemi *et al.*, "Comparison of different corneal imaging modalities using artificial intelligence for diagnosis of keratoconus: a systematic review and meta-analysis," *Eye and Vision*, vol. 11, May 2024, doi: 10.1186/s40662-024-00386-1.

[6] M. Al-Shakhsi *et al.*, "Advances in machine learning for keratoconus diagnosis," *International Ophthalmology*, vol. 45, no. 1, p. 128, May 2025, doi: 10.1007/s10792-025-02758-0.

[7] Z. J. Muhsin *et al.*, "Advances in machine learning for keratoconus diagnosis," *Int. Ophthalmol.*, vol. 45, no. 1, p. 128, Mar. 2025, doi: 10.1007/s10792-025-03496-4.

[8] S. Kumar *et al.*, "Explainable XceptionNet-Based Keratoconus Detection Using Infrared Images and Edge Computing," *IEEE Sensors Lett.*, vol. 8, no. 1, pp. 1–4, Jan. 2024, doi: 10.1109/LSENS.2023.3345472.

[9] *TensorFlow*, Google Brain Team, 2015. [Online]. Available: https://www.tensorflow.org

[10] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in *Proc. 36th Int. Conf. Mach. Learn. (ICML)*, 2019, pp. 6105–6114. [Online]. Available: https://arxiv.org/abs/1905.11946

[11] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Mach. Learn.*, vol. 63, no. 1, pp. 3–42, Apr. 2006, doi: 10.1007/s10994-006-6226-1.

[12] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2017, pp. 618–626, doi: 10.1109/ICCV.2017.74.

[13] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in *Proc. 36th Int. Conf. Mach. Learn. (ICML)*, 2019, pp. 6105–6114.

APPENDIX A (USER MANUAL AND TRAINING GUIDE)

The Appendix provides a full operational process flow of the Keratoconus Early Detection System (KEDS) or CorneaInsight. It includes the operation that should be performed by all involved party to ensure data security and accurate interpretation.

1. For Doctors: Interpreting Grad-CAM Heatmaps

The Visual Rationale Engine serves as the core component of KEDS, since its purpose is to win the physician's confidence in the system through visual explanation. Below are the guidelines on how to interpret the outputs of the AI model:

- **Visualizing Decision Regions:**

  For KEDS, the areas highlighted in these heatmaps are those that the EfficientNet-B0 consider while making a keratoconus diagnosis. They may include the area of protrusion, thinning, and asymmetry.

- **Identifying Pathological Zones:**

  Through this, the ophthalmologist will know that the AI is indeed focused on real clinical characteristic, not some imaging artifact.

- **Validating the "Smart Second Opinion":**

  By reviewing these highlight, the ophthalmologist can verify that the AI is focusing on clinically relevant morphological changes rather than image artifacts.

- **Synthesizing Multi-modal Data:**

  The physician must also combine this with the ExtraTreesClassifier feature importance ranking..

2. For Technicians: Topography Map Upload Process

Technicians handle the administrative and data-entry phase of the clinical workflow.

- **Supported File Formats:**

  All corneal topography map must be uploaded in JPEG or PNG formats.

- **Multi-modal Input:**

  Technicians must uploaded up to seven types of corneal maps, including elevation and curvature maps captured by imaging devices.

- **Preprocessing Automation:**

  Once uploaded, the backend API automatically standardized these images into 224*224*3 tensore for the deep learning core.

- **Input Validation:**

The system perform on-the-go validation to ensure numerical clinical parameter (like K_{max} and thinnest pachymetry) are within acceptable clinical bounds before the scan is processed.

3. For Patients: Accessing the Patient Portal

Patients are granted restricted access to monitor there longitudinal health status.

- **Secure Authentication:**
  The patient will utilize the Supabase Auth module. This is due to the fact that it supports safe authentication through login credentials such as email and password.

- **Monitoring Progression:**
  There will be a Progression Monitoring Chart available online. This chart shows the progress trends of the patient over various visitations.

- **Downloading Diagnostic Reports:**
  The patient is able to retrieve their diagnosis results. They can download the PDF file containing their diagnosis, confidence levels, and visual reasoning.

4. System Setup: Hardware and Performance Requirements

To achieve optimal performance and meet clinical safety standards, the following setup is required:

- **Inference Hardware:**
  An existing computer would be necessary in this regard. Ideally, the inference computer should be one that comes with an NVIDIA GPU featuring 4GB of VRAM.

- **Asynchronous Backend:**
  This was optimized to allow for parallel processing, and could handle as many as 20 simultaneous requests.

- **Processing Speed:**
  We were able to make sure that the entire hybrid diagnosis process takes no more than five seconds, thanks to our software-hardware combination.

- **Network Security:**
  In order to transfer PHI to the Supabase cloud database, the device would need to have access to the internet using TLS 1.3.

APPENDIX B (TECHNICAL DOCUMENTATION)

This Appendix contains the specifications regarding the infrastructure of the CorneaInsight (KEDS) backend architecture and its core's performance. The documentation are intended for system administrators and developers who would like to assure the technological integrity and clinical effectiveness of the software.

1. API Documentation (FastAPI Gateway)

CorneaInsight (KEDS) is based on a Python FastAPI gateway that processes asynchronous requests and makes machine learning inferences. All connections are encrypted by HTTPS/TLS.

1.1. Authentication Routes (/auth) (FastAPI Gateway)

CorneaInsight (KEDS) utilizes Supabase Auth and JWT tokens (Access and Refresh) to provide a high level of clinical data protection.

- **Register User (POST /auth/register):**
  The function allows for creating an account. It requires the assignment of particular roles (Ophthalmologist, Technician, Patient) for account users.

- **Login (POST /auth/login):**
  Authenticates credentials and issues a temporary Access Token for API calls and a permanent Refresh Token to maintain sessions.

- **Refresh Token (POST /auth/refresh):**
  Retrieves a new access token without requiring the user to re-enter credentials.

1.2. Prediction Routes (/predict)

This route initializes the Hybrid AI Core, which runs both EfficientNet-B0 image model and ExtraTreesClassifier numerical model.

1.2.1. Image-Based Predictions

- **Individual Maps (POST /predict):**
  Seven distinct corneal topography map, including elevation and curvature maps, among others, in JPEG format are uploaded to the server by the technician.

- **Composite View (POST /predict-composite):**
  A Belin/Ambrósio combined image can be uploaded to the server. The image is later segmented to distinct maps.

- **Output:**
  Probability-based predictions such as "Normal" or "Keratoconus." Moreover, a heat map known as Grad-CAM that emphasizes some zones in the images is also returned.

1.2.2. Numerical Parameter Scan (Post/parameter-scan)

Diagnosis is done base on some numerical indicators such as Kmax, ISV, and IVA.

- **Use Case:** Designed for model testing, validation, or clinical cases where full image sets are unavailable.

1.3. History and Management

It needs an Authorization Header with a Bearer Token to make sure that the data access is HIPAA-compliant.

- **Get User's Predictions (GET /predictions):**
  It shows previous records that will allow doctors to track the development of the disease.

- **Specific Prediction (GET /predictions/id):**
  It show all the detail about the diagnosis including the Confidence Score and the Visual Explanation (XAI).

- **Delete Prediction (DELETE /predictions/id):**
  Physicians who have the ability to edit or delete patient data are authorized to do so.

1.4. Technical Information

- **Processing Speed:**
  We have ensured that our backend can perform a full hybrid diagnosis in under 5 seconds.

- **Concurrency:**
  With FastAPI, we can concurrently process 20 API request at once.

- **Data Persistence:**
  The data is store permanently and securely within the Supabase (PostgreSQL) cloud database.

- **Documentation:**
  Our dynamic Swagger UI documentation can be accessed at the /docs endpoint.

APPENDIX C (AI MODEL BENCHMARKING AND INITIAL RESULTS)

The appendix discusses how the efficiency of the diagnostic core (CorneaInsight (KEDS)) was experimentally verify. The data obtained during the training and testing processes for each model are listed below.

1. Image-Based Model Performance (EfficientNet-B0)

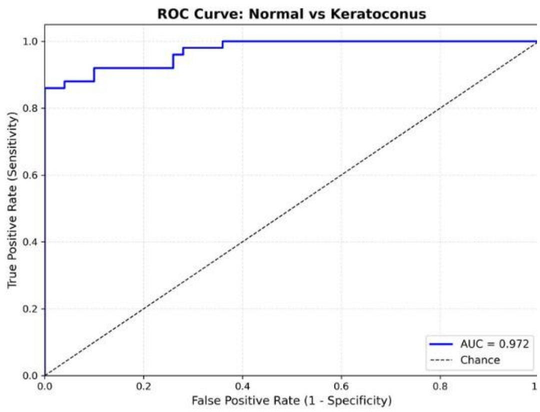
An image-based model in the diagnostic system is based on the architecture of the EfficientNet-B0 model. The model find disease pattern in the corneal topography map.

- **Diagnostic Accuracy:**

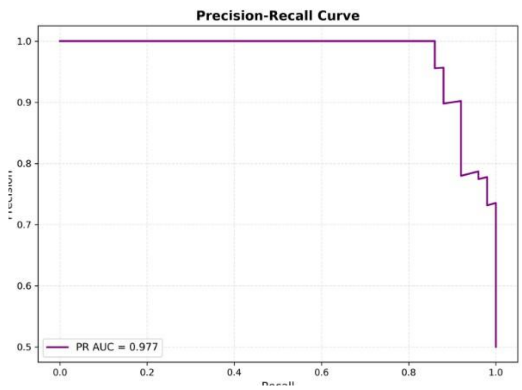
  The efficiency of the model was measured independently. The accuracy rate of the model was 91.0%

- **AUC-ROC Analysis:**

  The area under the curve (AUC) value was 0.972.

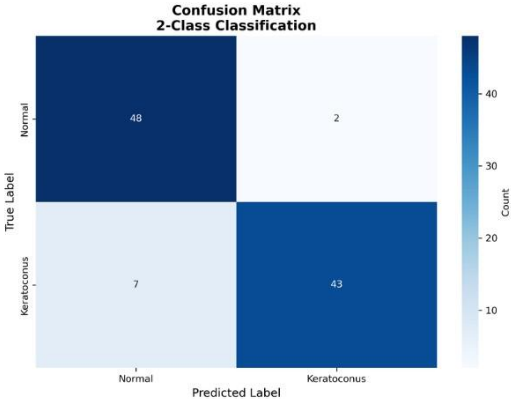


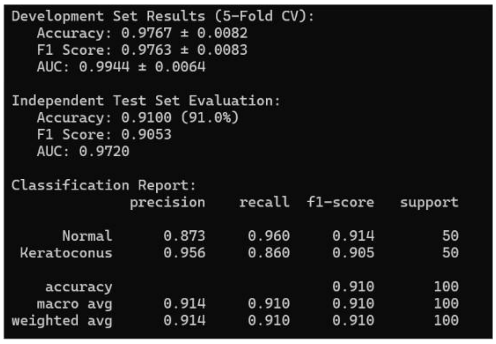
- **Precision and Recall:**
  - **Normal**: Precision of 0.873 and Recall of 0.960.
  - **Keratoconus:** Precision of 0.956 and Recall of 0.860.



- **Confusion Matrix:**

  In a test sample of 100 cases (50 Normal, 50 Keratoconus), the model correctly identified 48 normal eye and 43 keratoconus case.





```
Development Set Results (5-Fold CV):
   Accuracy: 0.9767 ± 0.0082
   F1 Score: 0.9763 ± 0.0083
   AUC: 0.9944 ± 0.0064

Independent Test Set Evaluation:
   Accuracy: 0.9100 (91.0%)
   F1 Score: 0.9053
   AUC: 0.9720

Classification Report:
              precision    recall  f1-score   support

      Normal      0.873     0.960     0.914        50
 Keratoconus      0.956     0.860     0.905        50

    accuracy                          0.910       100
   macro avg      0.914     0.910     0.910       100
weighted avg      0.914     0.910     0.910       100
```

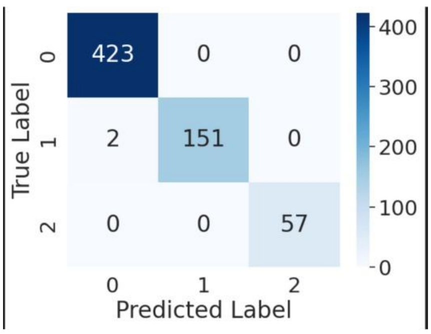
2. Numerical Model Performance (ExtraTrees & SVM)

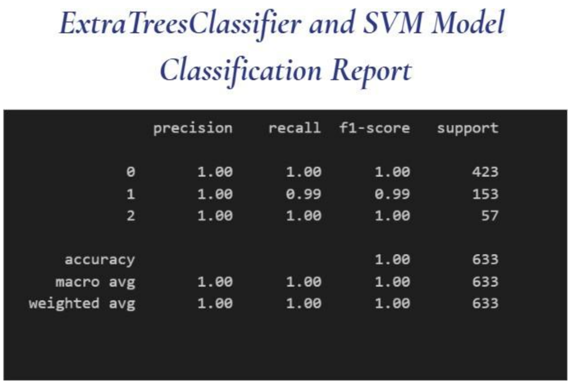
The numerical model makes use of structured medical information, including Kmax and corneal thickness.

- **Overall Accuracy:**
  The numerical model are able to score perfectly by achieving an accuracy score of 1.00 (100%) for the test sample set.

- **Classification Performance:**
  The numerical model had a score of 1.00 precision, recall, and F1-score for all the class samples.

- **Test Sample Distribution:**
  The test involved 633 sample, out of which 423 were class 0, 153 were class 1, and 57 were class 2 samples.






```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       423
           1       1.00      0.99      0.99       153
           2       1.00      1.00      1.00        57

    accuracy                           1.00       633
   macro avg       1.00      1.00      1.00       633
weighted avg       1.00      1.00      1.00       633
```

3. Summary of Benchmarking Outcomes

The integration of these two high-performing model into the Hybrid AI Core ensures a "smart second opinion" that is both rapid and reliable.

- **Fusion Stability:**

  The ensemble approach compensate for potential weaknesses in single data modality, such as poor image quality or missing clinical indices.

- **Processing Efficiency:**

  Clinical testing confirm that the system completes these complex multi-modal inferences in an average of 4.2 seconds.

- **Clinical Trust:**

  Every prediction is supported by the Visual Rational Engine (Grad-CAM), allowing doctors to visually verify the benchmarked results on a case-by-case basis.

APPENDIX D (PLAGIARISM REPORT)

*% detected as AI

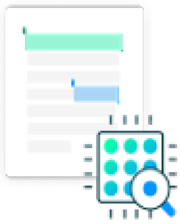
AI detection includes the possibility of false positives. Although some text in this submission is likely AI generated, scores below the 20% threshold are not surfaced because they have a higher likelihood of false positives.

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Frequently Asked Questions

How should I interpret Turnitin's AI writing percentage and false positives?
The percentage shown in the AI writing report is the amount of qualifying text within the submission that Turnitin's AI writing detection model determines was either likely AI-generated text from a large-language model or likely AI-generated text that was likely revised using an AI paraphrase tool or word spinner.



False positives (incorrectly flagging human-written text as AI-generated) are a possibility in AI models.

AI detection scores under 20%, which we do not surface in new reports, have a higher likelihood of false positives. To reduce the likelihood of misinterpretation, no score or highlights are attributed and are indicated with an asterisk in the report (*%).

The AI writing percentage should not be the sole basis to determine whether misconduct has occurred. The reviewer/instructor should use the percentage as a means to start a formative conversation with their student and/or use it to examine the submitted assignment in accordance with their school's policies.

What does 'qualifying text' mean?
Our model only processes qualifying text in the form of long-form writing. Long-form writing means individual sentences contained in paragraphs that make up a longer piece of written work, such as an essay, a dissertation, or an article, etc. Qualifying text that has been determined to be likely AI-generated will be highlighted in cyan in the submission, and likely AI-generated and then likely AI-paraphrased will be highlighted purple.

Non-qualifying text, such as bullet points, annotated bibliographies, etc., will not be processed and can create disparity between the submission highlights and the percentage shown.

FYP COMPLETION CERTIFICATE

Student Name: M Usman Ali Enrolment No: 01-131222-038
Student Name: Adil Shahzad Enrolment No: 01-131222-007
Programme of Study: Bachelor of Software Engineering
Project Title: CorneaInsight

It is to certify that the above students' project has been completed to my satisfaction and to my belief, its standard is appropriate for submission for evaluation. I have also conducted plagiarism test of this thesis using HEC prescribed software and found similarity index at ___________ that is within the permissible limit set by the HEC. I have also found the thesis in a format recognized by the department.

Supervisor's Signature: ____

Date: 15 April 2026 Name: Engr. Aamir Sohail

BS SE (Ked documents Human)

ORIGINALITY REPORT

8%	6%	2%	6%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Higher Education Commission Pakistan Student Paper	2%
2	www.coursehero.com Internet Source	1%
3	bahria.edu.pk Internet Source	1%
4	prr.hec.gov.pk Internet Source	<1%
5	dspace.cvut.cz Internet Source	<1%
6	Submitted to University of Wolverhampton Student Paper	<1%
7	Submitted to University of SWAT Student Paper	<1%
8	Submitted to University of Northumbria at Newcastle Student Paper	<1%
9	Submitted to MAHSA University Student Paper	<1%
10	senior.ceng.metu.edu.tr Internet Source	<1%

11	Submitted to Asia Pacific University College of Technology and Innovation (UCTI) Student Paper	<1 %
12	Submitted to University of Bahrain Student Paper	<1 %
13	Submitted to Kingston University Student Paper	<1 %
14	Submitted to University of Central England in Birmingham Student Paper	<1 %
15	Submitted to City University of Hong Kong Student Paper	<1 %
16	journal2.iicet.org Internet Source	<1 %
17	Submitted to Arab Open University Student Paper	<1 %
18	Submitted to Laureate Higher Education Group Student Paper	<1 %
19	opensourcevsproprietary.blogspot.com Internet Source	<1 %
20	B. G. Nagaraja, S. Kannadhasan. "Information and Communication Systems", CRC Press, 2026 Publication	<1 %
21	Submitted to Kean University Student Paper	<1 %
22	Submitted to University of Huddersfield Student Paper	<1 %

23	Submitted to lahore garrison LGU Student Paper	<1%
24	repository.uindatokarama.ac.id Internet Source	<1%
25	Submitted to Bahcesehir University Student Paper	<1%
26	comum.rcaap.pt Internet Source	<1%
27	repositorio.uam.es Internet Source	<1%
28	www.wattpad.com Internet Source	<1%
29	inspire.cse.unt.edu Internet Source	<1%
30	restaurantvisio.fr Internet Source	<1%
31	Ronald Giachetti. "Design of Enterprise Systems - Theory, Architecture, and Methods", CRC Press, 2019 Publication	<1%
32	Ton Duc Thang University Publication	<1%
33	Xue, Zhaokun. "Discovery and Recommendation Systems for Scientific Data", Tufts University, 2021 Publication	<1%
34	download.bibis.ir Internet Source	<1%
35	www.advantronic.es	

Internet Source

<1%

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		